

(19)



Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 0 738 957 A2

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:
23.10.1996 Bulletin 1996/43

(51) Int. Cl.⁶: G06F 3/12, G03G 15/00,
G06K 15/00

(21) Application number: 95116577.8

(22) Date of filing: 20.10.1995

(84) Designated Contracting States:
DE FR GB

(30) Priority: 18.04.1995 JP 92615/95
19.04.1995 JP 93818/95
19.04.1995 JP 93819/95
19.04.1995 JP 93820/95
19.04.1995 JP 117982/95
20.04.1995 JP 95447/95
20.04.1995 JP 95448/95
20.04.1995 JP 95449/95

(71) Applicant: FUJI XEROX CO., LTD.
Minato-ku Tokyo 107 (JP)

(72) Inventors:
• Suzuki, Akihiro,
c/o Fuji Xerox Co., Ltd.
Takatsu-ku, Kawasaki-shi, Kanagawa (JP)

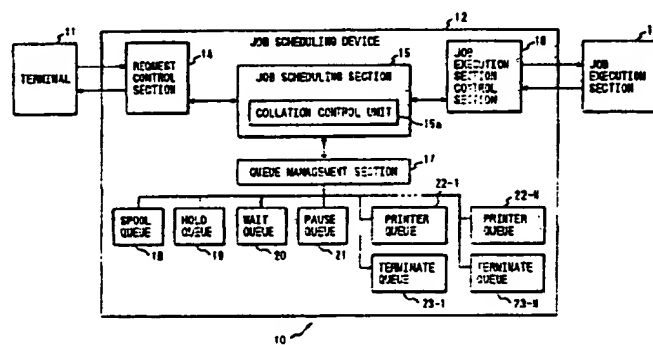
• Yamada, Kentaro,
c/o Fuji Xerox Co., Ltd.
Takatsu-ku, Kawasaki-shi, Kanagawa (JP)
• Nishiyama, Koji,
c/o Fuji Xerox Co., Ltd.
Takatsu-ku, Kawasaki-shi, Kanagawa (JP)
• Nakatani, Tooru,
c/o Fuji Xerox Co., Ltd.
Takatsu-ku, Kawasaki-shi, Kanagawa (JP)
• Nakamura, Yoh,
c/o Fuji Xerox Co., Ltd.
Takatsu-ku, Kawasaki-shi, Kanagawa (JP)

(74) Representative: Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München (DE)

(54) Job scheduling system for print processing

(57) A user equipment issues a job for a plurality of documents to a printing system by one operation. The user equipment assigns attributes to the job, the attributes including the specification of collation/uncollation, the designation of processing start instruction wait, the designation of exclusive processing, and the designation of password input wait. The printing system effects printing of the plurality of documents corresponding to attributes of the documents under control of a job scheduling device. The job scheduling device carries out the pausing of documents included in the job, the modification of attributes, addressing of document receiving failures, and search of a document whose format is to be converted, thereby making it possible to efficiently print the plurality of documents.

FIG. 1



EP 0 738 957 A2

informs the job scheduling device of the completion of the processing of that document. The job scheduling device considers the processing of the document for which the acknowledgement has been issued to have been completed, and the job is moved to a terminate (completion) queue (not shown) after the processing of the final document of the job has been completed. The job scheduling device issues a processing request to the job execution section 6 with respect to the next job 2 at the head end of the printer queue 5.

In some case, the job scheduling device cannot issue a document processing request to the job execution section 6 even when the job execution section 6 is liable to carry out processing, which results in an unprocessed document is in the printer queue 5.

For example, Fig. 21 shows that processing requests for all jobs have already been issued with respect to the job 1, and that a job 1-4, i.e., a document 4 of the job 1, is being processed in the job execution section 6. When the job 1-4 is shifted to step 2 after having been subjected to the processing at step 1, it becomes possible to carry out the processing of step 1. The job execution section 6 informs the job scheduling device that it can accept processing. However, the job scheduling device has not yet received the acknowledgement of the completion of the processing of jobs 1-2, 1-3, and 1-4 from the job execution section 6. Accordingly, the job scheduling device can neither shift the job 1 from the head end of the printer queue 5 nor issue a processing request for a job 2-1. In other words, in spite of the fact that the job execution section 6 informs the job scheduling device that it can accept processing, the job execution section 6 cannot receive a processing request for the job 2-1 from the job scheduling device until the processing of the job 1-4 is finished. Hence, the job execution section 6 is idle for a waiting time, and it takes a long time to execute job processing. As a result of this, even if the job execution section 6 can process a plurality of documents at one time, it becomes impossible to effectively utilize the throughput of the job execution section.

As previously mentioned, the job scheduling device controls jobs by checking whether or not a job at the head end of the printer queue 5 is set to the processing start wait, or the processing completion wait, and the password input wait. If processing requests for documents were issued with respect to jobs other than the job at the head end of the queue when the job scheduling device received the acknowledgement from the job execution section 6, it will become impossible for the job scheduling device to control these jobs.

Another object of the present invention is to provide a job processing system capable of continuously issuing document processing requests without obstructing job control such as a processing start wait.

When print processing is conventionally carried out using a print server connected to a network system, it is necessary to send the print server print data described in a print format which a printer to be connected to the

server can interpret. For this reason, a local user in the network system has to send print data after confirming the printer format of a printer which accepts a print processing request. If the print data were sent together with a wrong print format for that printer, the print data might not be printed.

A print server is put forward which converts the print format of received print data to another print format which a printer connected to the print server can interpret when the print data described in a print format which the printer cannot interpret. The print format is a PDL format such as PostScript or Interpress.

Japanese Patent Unexamined Publication No. Hei. 3-251915 discloses a print server, wherein a printer name corresponding to a print command is sent to a print server before the transmission of print data, the print server selects print command converter means corresponding to the received print command, and the thus selected print command converter means control a printer connected to the print server.

However, in the case of the above mentioned conventional print server, the print format of the print data is converted immediately before the printer processes the print data. Hence, the printer comes to a standstill during the conversion of the print format, which in turn deteriorates the throughput of the printer. Particularly in the case of a completion type job, the conversion of a print format is not carried out until all documents are received, and hence the printer stops for a longer period of time.

Similarly, the print server disclosed in Japanese Patent Unexamined Publication No. Hei. 3-251915 converts the print format of print data to a data format of a printer connected to the server immediately before print processing is carried out. Therefore, the printer has to stop during the conversion of the print format.

Still another object of the present invention is to provide a print processing device designed to reduce the idle time of a printer and improve the throughput of the printer by previously converting a job including documents which need conversion processing.

In the scheduling of a conventional one-job multi-document, when it is necessary to process a plurality of documents as one job, an instruction to that effect is sent from terminal equipment together with a schedule attribute. This schedule attribute basically comprises two types of attribute; namely, an after-complete attribute and a before-complete attribute. The after-complete attribute means that a print processing device carries out scheduling in such a way as to execute a job upon receipt of all print documents related to that job. The before-complete attribute means that a print processing device carries out scheduling in such a way as to execute a job with respect to a received print document every time receiving each of print documents of that job.

By the way, document data to be transmitted might be interrupted as a result of a network failure while print requests for a plurality of print documents are being

before the occurrence of the failure. For example, on the assumption that a plurality of print jobs are received from a plurality of terminals through a network, and that a job scheduling device which prints the plurality of print jobs one after another using a job execution section (a printer) is used, if any failures occurred in the job scheduling device or the printer, the state of the job scheduling device or the printer will be temporarily stored in a log file, and thereafter attempts to recover the job scheduling device or the printer from the failure will be made based on an analysis of the failures by a system manager.

In other words, once such failures have occurred, the failures affect each of the jobs held in the job scheduling device, and therefore it often takes a long time to recover the jobs from the influence of the failures. For example, if a failure arises during the course of deletion of a job held in the job scheduling device, print data relating to that job will be deleted, but attributes of that print data will remain undeleted. For this reason, the system manager has to delete the attributes later.

For these reasons, a degree of improvement in an operating efficiency attained as a result of reducing the influence of failures exerted on a job if the above mentioned failures arise, becomes important. Particularly, in a system which carries out parallel processing such as multiprocessing, one failure affects all other portions in the system.

The object of the present invention is to solve the above mentioned drawbacks in the conventional technology and to provide a job scheduling device capable of reducing the influence of system failures if they arise in a system and executing job processing according to an instruction from a user.

A conventional printing system, such as a so-called print system and a copying system, employs a known technology in which print attributes (hereinafter referred to as attribute information) consisting of a paper size, a tray number, and the availability of double-sided printing are added to print data, and printing is carried out based on the attribute information. For example, when print data or copying data are output, attribute information consisting of, for example, a paper size, the number of pages, and the availability of double-sided printing, necessary for printing is set. As a result of sending the attribute information to an output device together with document data or copying data, the print data or the copying data are output in a desired format. In this way, when a print request is issued, printing is carried out based on the attribute information, and therefore the handling of the attribute information becomes important.

However, according to such a conventional technology, the user who issues a print request or a terminal equipment which the user uses, added the attribute information to print data. For this reason, if a print server, or the like, accepted the print request as a print job, it would not have been allowed to change the attribute information. Hence, if the user erroneously

specified the attribute information or if the terminal added the attribute information to the print data because the user had forgot to specify the attribute information, it would have become necessary to temporarily cancel the print job once and to issue a new print request given desired attribute information. This imposes an increased burden on the user, as a result of the re-registration of a print request, and brings about a drop in the efficiency of use of a network. Further, a time lag arises before a print output is obtained. Namely, the increased burden on the user, a drop in the efficiency of use of the network, and a print waiting time becomes important.

The present invention is conceived to solve the above mentioned drawbacks in the conventional technology, and the object of the invention is to provide a job scheduling device capable of easily modifying attribute information by reducing a user's burden, a drop in the efficiency of the use of a network, and a print waiting time.

SUMMARY OF THE INVENTION

A first object of the present invention is to provide a job processing system capable of improving the processing efficiency of a job when a plurality of jobs having the same contents are output.

A second object of the present invention is to provide a job processing system designed so as to be able to continuously output document processing requests without obstructing job control such as a processing start wait.

A third object of the present invention is to provide a print processing equipment designed so as to reduce idle time of a printer and to improve a throughput of the printer by previously converting a job including documents which need conversion.

A fourth object of the present invention is to provide a job processing system which prevents the interruption of processing for a long period of time and the occupying of a printer for a long period of time even if the processing system cannot accept the next document of a job for reasons of failures.

A fifth object of the present invention is to provide a job processing equipment capable of solving the previously mentioned drawbacks in the conventional technology, modifying attributes of a job retained in a queue, easily cancelling a specific document included in a one-job multidocument, and improving processing efficiency by effecting the interruption and resumption of processing on a job-by-job basis.

A sixth object of the present invention is to provide a job scheduling device capable of reducing the influence of system failures and carrying out job processing according to a user's instruction even when failures arise in a system.

A seventh object of the present invention is to provide a job scheduling device capable of easily modifying attribute information by reducing a user's burden, a drop

document; and control state setting means which, if the processing completion wait is set to information which specifies the leading document of the job stored in the queuing means, renders that job in a processing completion wait state, and wherein the job scheduling device sequentially retrieves jobs stored in the queuing means when the job execution section becomes enabled to accept processing, issues a processing request for a corresponding document when there is information specifying a document to which a processing request can be issued, and when a job is placed in the processing completion wait state, prevents the issue of processing requests with respect to a document for that job and documents for subsequent jobs until that job is released from the processing completion wait state by a user's instruction or a timeout.

According to a fifth aspect of the present invention, there is provided a job processing system comprising a terminal equipment for issuing a processing request by handling a plurality of documents as one job, a job execution section for printing the documents, and a job scheduling device which accepts a document input from the terminal equipment through a network and issues a processing request relating to that document to the job execution section, the terminal equipment comprising: control information setting means for specifying a password input wait for a leading document among the plurality of documents, and the job scheduling device comprising: preparation means for preparing information which specifies a received document; queuing means for storing the information which specifies the document by associating the information on a job-by-job basis; control information setting means which, if a password input wait is set for the leading document among a plurality of received documents, sets the password input wait to information which specifies that leading document; and control state setting means which, if the password input wait state is set to information which specifies the leading document of the job stored in the queuing means, renders that job in a password input wait state, wherein the job scheduling device sequentially retrieves jobs stored in the queuing means when the job execution section becomes enabled to accept processing, issues a processing request for a corresponding document when there is information specifying a document to which a processing request can be issued, and when a job is placed in the password input wait state, prevents the issue of processing requests with respect to a document of that job and documents of subsequent jobs until that job is released from the password input wait state by a user's instruction or a timeout.

According to a sixth aspect of the present invention, there is provided a job processing system comprising a terminal equipment for issuing a processing request, and a job scheduling device which sequentially processes jobs by storing the jobs received from the terminal equipment in a queue and issuing a processing request, relating to a document specified by the job stored in the queue, to a job execution section, the ter-

terminal equipment comprising: attribute information adding means for adding information relating to job wait control and message information relating to the wait control to the job request as attribute information, and the scheduling device comprising: job information preparing means for preparing job information which specifies a received job; attribute information setting means for setting attribute information included in the received job in the job information; a queue for storing the prepared job information in order; control state setting means which, if wait control is set to the job information stored in the queue, renders a job associated with that job information in a wait control state when processing of that job is started or completed; and message information informing means which, when the job enters the wait control state, informs the terminal equipment of message information set with respect to that job.

According to a seventh aspect of the present invention, there is provided a print processor which prints a document in response to a job request received through a network, the print processor comprising: job accepting means for accepting the job request; queuing means for storing the accepted jobs in sequential order; output means for printing a document specified by the job stored in the queuing means; converting means for converting the document into a format which the output means can interpret; and conversion control means for causing the converting means to convert a document which needs to be converted when being printed by the output means, wherein the converting means informs the conversion control means that it can carry out conversion when conversion processing becomes available, and wherein the conversion control means retrieves a document in a format which the output means cannot interpret from among the documents specified by the job stored in the queuing means, and causes the converting means to convert the format of that document.

According to an eighth aspect of the present invention, there is provided a print processor which carries out printing upon receipt of a job processing request, including an instruction for processing print requests for a plurality of documents as one job and scheduling attributes, from a plurality of terminals through a network, the print processor comprising: timer means which, upon receipt of a job processing request from the terminal, checks whether or not a next print document was received within a predetermined period of time by monitoring a series of documents included in the job processing request; scheduling attribute judging means which, upon receipt of the job processing request, judges whether the scheduling attributes are after-complete attributes which carry out scheduling in such a way that the execution of a job is started upon receipt of all print documents related to the job or before-complete attributes which carry out scheduling in such a way that a job is executed with respect to a received print document every time each print document is received; final document processing means which, when the timer means has judged that the next print document could

Fig. 8 is a schematic representation showing queue objects stored in a printer queue and information retained therein (uncollated);

Fig. 9 is an explanatory view showing output results when a job is output in an uncollated manner in the first embodiment;

Fig. 10 is a schematic representation showing queue objects in a second embodiment of the present invention;

Fig. 11 is flowchart showing processing procedures when a document is accepted in the second embodiment;

Fig. 12 is a flowchart showing processing procedures when document processing requests are continuously issued in the second embodiment;

Fig. 13 is a flowchart showing the continuous issue of document processing procedures, and processing procedures when a job processing start wait is controlled in the second embodiment;

Fig. 14 is a flowchart showing the continuous issue of document processing procedures, and processing procedures when a job processing completion wait is controlled in the second embodiment;

Fig. 15 is a flowchart showing the flow of acceptance of a processing completion acknowledgement from a job execution section in the second embodiment;

Fig. 16 is a flowchart showing the continuous issue of document processing procedures, and processing procedures when a password input wait is controlled in the second embodiment;

Fig. 17 is an explanatory view showing the relationship between a printer queue and a terminate queue in which jobs waiting for processing start are queued in the second embodiment;

Fig. 18 is an explanatory view showing the relationship between a printer queue and a terminate queue in which jobs waiting for processing completion are queued in the second embodiment (part 1);

Fig. 19 is an explanatory view showing the relationship between a printer queue and a terminate queue in which jobs waiting for processing completion are queued according to the second embodiment (part 2);

Fig. 20 is an explanatory view showing the relationship between a printer queue, a terminate queue, and a wait queue in which jobs waiting for password input are queued in the second embodiment;

Fig. 21 is an explanatory view showing the relationship between the state of a printer queue and a document to be processed in a job execution section in the second embodiment;

Fig. 22 is a block diagram showing the detailed configuration of a job scheduling section in a third embodiment of the present invention;

Fig. 23 is a block diagram showing the overview of a print processing system in the third embodiment;

Fig. 24 is a flowchart showing the flow of processing from the receipt of a document to the queuing of

the document in a printer queue in the third embodiment;

Fig. 25 is a flowchart showing the flow of conversion processing when a conversion section becomes possible to process in the third embodiment;

Fig. 26 is a functional block diagram showing one example of configuration of a job scheduling section used in a fourth embodiment of the present invention;

Fig. 27 is a block diagram showing one example of the configuration of a print processing device of the fourth embodiment;

Fig. 28 is a block diagram showing one example of a job control elementary section in the fourth embodiment;

Fig. 29 is a block diagram showing one example of a queuing management section in the fourth embodiment;

Figs. 30a to 30c are schematic representations showing one example of data structure of a document printing request in the fourth embodiment;

Fig. 31 is a flowchart showing the flow of processing of a job scheduling section in the fourth embodiment;

Fig. 32 is a flowchart showing the flow of timeout processing at the time of after-complete processing in the fourth embodiment;

Fig. 33 is a flowchart showing the flow of timeout processing at the time of before-complete processing in the fourth embodiment;

Fig. 34 is a block diagram showing the configuration of a job control elementary section in a fifth embodiment of the present invention;

Fig. 35 is a schematic representation showing the configuration of a job processing system of the fifth embodiment;

Fig. 36 is a schematic diagram showing the configuration of a pausing section shown in Fig. 34;

Fig. 37 is a flowchart showing processing procedures carried out by the pausing section shown in Fig. 36;

Fig. 38 is a schematic representation showing the configuration of a resuming section shown in Fig. 34;

Fig. 39 is a flowchart showing processing procedures executed by the resuming section shown in Fig. 38;

Figs. 40a to 40c are schematic representations showing one example of the states of a printer queue and a pause queue when pausing processing and resuming processing are executed in the fifth embodiment;

Fig. 41 is a view showing a processing sequence between the job scheduling section and the job execution section shown in Fig. 34;

Fig. 42 is a block diagram showing the overall configuration of a job processing system and the detailed configuration of a job scheduling device

value of this request counter. The counter decrements from the initial value by one every time a processing request is issued.

Complete Count: Processing completion count which represents the number of acknowledgements of processing completion output from the job execution section 13. This counter is set to zero when document information of the current job information is prepared.

Complete: Final document identifier which represents the final document when TRUE.

These queue objects are stored in predetermined queues depending on the state of the job. In other words, a queue object of a corresponding job is stored in a queue. Practical documents are stored in memory (not shown) associated with queues. Throughout the following description, queue objects stored in queues are simply called documents.

The job scheduling section 15 transfers the job delivered from the request control section 14 to a queue management section 17, and queues corresponding to the job execution section 13 and queues necessary for scheduling are managed using the queue management section 17. Scheduling is carried out allowing for an assignment and processing conditions of the job execution section 13, and a print processing request is issued to the job execution section 13 by transferring a job stored in the queue management section 17 to the job execution section 13. A queue object of the job transferred from the request control section 14 is stored in a printer queue 22 if the job is a non-acceptance completion type sequential processing job; is stored in a spool queue 18 if the job is an acceptance completion type sequential processing job; and is stored in a hold queue 19 if the job is an unscheduled job. If another document follows that job request, information is set in the document information section of that document upon reference to job information of another job having the same job identifier, and that document is added to a queue object with the same job identifier using the queue management section 17.

Of the jobs stored in queues other than the hold queue 19 and a terminate queue 23, jobs including documents which need conversion are converted at the time when the job execution section 13 which carries out corresponding conversion becomes enabled.

A collation control unit 15a refers to job information and document information of the jobs stored in the printer queue 22 of the queue management section 17. If the collation identifier (Collate Flag) is set to TRUE (collation), a processing request issued to the job execution section 13 is controlled in such a way that the job is output in a specified number in a collated manner using the job information and the document information. If the collation identifier is set to FALSE (uncollation), the processing request issued to the job execution section 13 is controlled in such a way that the job is output in a specified number in an uncollated manner using the job information and the document information. Upon receipt of an acknowledgement representing that the

job execution section 13 can accept processing from the job execution section 13, the job scheduling section 15 sequentially fetches jobs from the leading end of the printer queue 22, and issues a document processing request to the job execution section 13 via the job execution section control section 16, with the use of the collation control unit 15a, in such a way that the job is output in a collated or uncollated manner as specified by a user. The job scheduling section 15 then receives a status acknowledgement of the document, to which the processing request was issued, from the job execution section 13 via the job execution section control section 16, and executes the scheduling of another job.

The job execution section control section 16 issues a request for print processing to the job execution section 13 by transferring the document processing request obtained from the job scheduling section 15 to the job execution section 13. Further, the job execution section control section 16 transfers a status acknowledgement relating to the job processing which was obtained from the job execution section 13 to the job scheduling section 15.

The queue management section 17 prepares various types of queue in compliance with an instruction from the job scheduling section 15, and stores a series of queue objects having the common job identifier as one job by associating them with each other. The spool queue 18 is a queue in which acceptance-completion type sequential processing jobs are stored, and a job is managed within this queue until all documents of that job are completely obtained. The hold queue 19 temporarily holds a document. If a job hold is specified for a job request, the job request is stored in this hold queue 19 regardless of whether the job is an acceptance-completion type sequential processing job or a nonacceptance-completion type sequential processing job. The job stored in the hold queue 19 is released from its held state according to the user's instruction. If the job is acceptance-completion type sequential processing job, the job is transferred to the spool queue 18. On the other hand, if the job is nonacceptance-completion type sequential processing job, the job is transferred to the printer queue 22. A wait queue 20 is a kind of processing wait queue. When a job waiting for a password input from the user has timed out, the job is transferred from the printer queue 22 to this queue 20. When a password is input by the user, the job returns to the printer queue 22. A pause queue 21 is a queue in which a job subjected to temporal interruption of processing is stored. If a system manager performs a predetermined operation through a server manager (not shown), the job moves to the pause queue 21 from the printer queue 22 or returns from this pause queue 21 to the printer queue 22. The printer queue 22 (1 to N) is provided corresponding to job outputs (not shown) of the job execution section 13. A job waiting for print processing is stored in this queue 22. A terminate queue 23 (1 to N) is provided corresponding to the printer queue 22 (1 to N). The job, the job processing of which was completed with respect to

Hence, if the processing request count in step S206 is not zero, the same job processing will be executed from the start. To avoid this, the processing request document identifier is reset to zero, and processing requests are again issued with respect to all of the documents. The processing is repeated until the processing request count becomes zero. It becomes possible to output a specified number of copies of the documents in order by carrying out the above processing in correspondence to the specified number of copies.

If the collation identifier in step S202 is not TRUE (collation is not available), the collation control unit 15 acquires document information with respect to a processing request document identifier (Current Doc Num) = 1 of the job identifier XX (step S208). On the assumption that the processing request count (Request Count) = processing request count - 1, a processing request is issued to the job execution section 13 through the job execution section control section 16 (step S209). Subsequently, it is judged whether or not the processing request count (Request Count) of the document information is zero (step S210). If the processing request count is not zero, the processing will return to step S209, and a processing request for the same document will be issued. On the other hand, if the processing request count is zero, it is judged whether or not the final document identifier (Complete) of this document information is TRUE (step S211). If the final document identifier is not TRUE, document information with respect to the processing request document identifier + 1 of the job identifier XX will be obtained (step S212), and the processing will return to step S209.

If the final document identifier is TRUE in step S211, the issue of the processing request of the job identifier XX will be terminated.

For example, if the number of copies of the job is two, the processing request count will also be set to two. If the processing request count in step S210 is not zero, a processing request will be again issued to the same document. This processing will be repeated with respect to all documents, whereby a specified number of copies of a document will be separately output for each document.

Processing completion procedures in the job scheduling section 15 when a job with a job identifier XX was output in collated or uncollated manner will be described with reference to a flowchart shown in Fig. 5.

Upon receipt of an acknowledgement of the completion of the processing of a job identifier XX (Job ID) from the job execution section 13 via the job execution section control section 16 (step S301), the job scheduling section 15 acquires document information of the document which was subjected to processing completion (step S303) as well as job information of the job identifier XX (step S302). On the assumption that a processing completion count (Complete Count) is +1 (step S304), it is judged whether or not the processing completion count is equal to a job copy number count (Job Copy Count) (step S305). If these two values are

not identical with each other, the processing will return to step S301. On the other hand, if these two values are equal to each other, the processing of that document will be terminated (step S306), and it is judged whether or not the final document identifier is TRUE (step S307). If the final document identifier is not TRUE, the processing will return to step S301. However, if it is TRUE, the processing of the job identifier XX will be terminated.

An explanation will then be given of a specific example where the non-acceptance completion type sequential job is output in collated or uncollated manner. Initially, an explanation will be given of the case where a one-job/two-document job is output in a collated manner with a job copy number 2.

Fig. 6 is a diagrammatic representation showing queue objects stored in a printer queue and information retained therein. Reference symbol j1 designates job information of an object job, d1 is document information of a document identifier (Doc Seq Num), and d2 is document information of a document identifier 2.

In an initial state represented by "1", a copy count (Copy Count) of the job information j1 is zero, and a processing request count (Request Count) of the document identifier (Doc Seq Num) 1 and a job copy number count (Job Copy Count) are equal to each other. This shows that a processing request for the first lap of the first document is not issued yet.

If the job execution section 13 becomes enabled, the collation control unit 15a of the job scheduling section 15 issues a processing request with respect to a document of a job with the document identifier 1 ("2"). At this time, the copy count (Copy Count) of the job information j1 is 1, and hence the job processing request count (Request Count) of the document information d1 becomes 1.

When the job execution section 13 becomes enabled, the collation unit 15a issues a processing request with respect to a document with a document identifier 2 ("3"). A processing request with respect to a document subsequent to the document with the document identifier 2 is issued when a processing request count of document information of this document is larger than the processing request count of the document with the document identifier 1. In "3", the processing request count of the document information d2 changes from 2 to 1, and therefore the processing request document identifier (Current Doc Num) = the document identifier (which is 2 in this example).

A processing request for the second lap of the document (the first document) with the document identifier 1 is issued when the processing request count of the final document (Complete Count = TRUE) is equal to the processing request count of the document (second document) with the document identifier 2. In other words, when the job execution section 13 becomes enabled after "3", the collation control unit 15a issues a processing request for the second lap of the document with the document identifier 1 ("4"). At this time, the copy count of the document identifier 1 becomes two,

Therefore, it becomes possible to improve job processing efficiency when a plurality of jobs having the same contents are output.

With reference to Fig. 1 and Figs. 22 to 25, a job processing system according to a second embodiment of the present invention will now be described. The job processing system of this embodiment is arranged in such a way that job control, such as a processing start wait, a processing completion wait, and a password input wait, is carried out even in the case of a job other than a job at the leading end of a queue when a one-job multidocument is scheduled. Since the overall configuration of the job processing system is the same as that of the first embodiment (shown in Fig. 1), and hence the explanation thereof will be omitted here for brevity.

In the second embodiment, the terminal 11 adds an undesigned identifier to the leading document among a plurality of documents previously prepared in this terminal. A job identifier delivered to the leading document from the job scheduling device 12 is added to the second and later documents, and the information which represents the end of the plurality of documents is added to the final document. Then, the plurality of documents are sent, and a processing start wait, a processing completion wait, and a password input wait are respectively set to the leading document of the plurality of documents. Further, message information may be set in accordance with the setting of the processing start wait, the processing completion wait, and the password input wait.

During the processing start wait, it is checked whether or not the document is set to a processing start wait when the processing of the job is started. If the document is set to the processing start wait, the processing of a job will not be started. The processing of that job is started when the processing start wait is cancelled by a start instruction from the user or a timeout. The terminal 11 adds the designation of processing start wait to the job request as information relating to the processing start wait control of the job. Further, a message, for example, "Waiting for processing start" or "Replace paper with OHP film" may be added to the job request as the message information relating to wait control.

During the processing completion wait, it is checked whether or not the document is set to a processing completion wait when the processing of the job was completed. If the document is set to the processing completion wait, the job will not be output. The output of that job is started when the processing completion wait is cancelled by a start instruction from the user or a timeout. The terminal 11 adds the instruction of processing completion wait to the job request as information relating to the processing completion wait control of the job. Further, a message, for example, "Waiting for processing completion" or "Replace tray" may be added to the job request as the message information relating to wait control.

During the password input wait, it is checked whether or not the document is set to a password input

wait when the processing of the job was completed. If the document is set to the password input wait, the processing of the job will not be started. The processing of that job is started only when a valid password is input. If a valid password is not input after the passage of a predetermined period of time, that job is transferred to another queue, and the processing of the following job moves up and is started. Further, when the password is input, the previous job is returned to the original queue, and the processing of that job is started subsequent to the job which is currently processed. The terminal 11 adds, the designation of password input wait and password data for use in judging whether or not the input password is valid, to the job request as information relating to the password input wait control of the job. Further, a message, for example, "Waiting for password input" or "Input password" may be added to the job request as the message information relating to wait control.

Information relating to the control wait of a job and the message information can be specified individually or in combination. For example, when a job is output in the form of an OHP film by manually feeding the film, a processing start wait and a processing completion wait are designated. Upon receipt of an acknowledgement of processing start wait, the user inserts paper during the processing of the job is interrupted. Thereafter, upon receipt of an acknowledgement of processing completion wait, the user attaches an ordinary tray to the printer while the printer is in an output wait state after the completion of the job processing.

The job scheduling device 12 is a module which schedules documents sent from the terminal 11 using queues, and sends the documents to the job execution section 13 for print processing. The job scheduling device 12 includes the request control section 14, the job scheduling section 15, the job execution section 16, and the queue management section 17.

The request control section 14 judges, upon receipt of a document including an unspecified identifier from the terminal 11, that the document is the leading document, and judges, upon receipt of a document including information which indicates the final, that the document is the final document among a plurality of documents. Further, the request control section 14 sets a job identifier which is an identifier common to the plurality of documents (from the leading document to the final document), and sends the job identifier back to the terminal 11.

Fig. 10 is a schematic representation showing queue objects which are prepared in the request control section 14. The request control section 14, upon receipt of the leading document, prepares a queue object (which stores information common to all of documents) of a job including the documents, and sets a job information section. In the job information section, information identifying the job (hereinafter referred to as job information) is prepared based on the information set to the leading document.

wait, and a password input wait while maintaining the order of jobs. Therefore, the job scheduling section 15 performs the following control with respect to jobs queued in the printer queue of the queue management section 17, that is, a job in a processing start wait state, a job in a processing completion wait state, and a job in a password input wait state.

(1) Job in processing start wait state

It is checked whether or not this job is set to a processing start wait before it is checked whether or not a processing request is issuable with respect to the leading document of that job. If the processing start wait is set in the job information section of queue objects, that job is rendered in a processing start wait state. No processing request is issued with respect to a document of this job and documents of subsequent jobs until the job is released from the processing start wait state by the user's instruction input from the terminal 11 or a timeout.

(2) Job in processing completion wait state

It is checked whether or not the job is set to a processing completion wait before it is checked whether or not a processing request is issuable to the leading document of that job. If the processing completion wait is set in the job information section of queue objects, a job identifier of this job is set to a printer queue. In this case, the job identifier is set in such a way that each printer queue retains a job identifier of the job which is in a job completion wait state. Hereafter, a processing request is issuable only for this job. After the completion of the processing of all documents of this job, this job is rendered in a processing completion wait state. No processing request is issued with respect to documents of subsequent jobs until this current job is released from the processing completion state by a user's instruction input from the terminal 11 or a timeout.

(3) Job in password input wait state

It is checked whether or not the job is set to a password input wait before it is checked whether or not a processing request is issuable to the leading document of that job. If the password input wait is set in the job information section of queue objects, this job is rendered in a password input wait state. No processing request is issued with respect to a document of this job and documents of subsequent jobs until this current job is released from the processing completion state by a user's instruction input from the terminal 11 or a timeout.

The job execution section control section 16 issues a request for print processing to the job execution section 13 by moving the processing request from the job scheduling section 15 to the job execution section 13. Further, the job execution section control section 16

transfers a status acknowledgement relating to the job delivered from the job execution section 13 to the job scheduling section 15.

The queue management section 17 prepares various types of queue in accordance with an instruction from the job scheduling section 15, and queues a series of queue objects having a common job identifier as one job.

The spool queue 18 queues documents of a completion type job, and manages the job until all documents of that job are completely obtained. The completion type job is characterized in that print processing is not started until all documents which form a job are completely obtained. On the other hand, a noncompletion type job is characterized in that print processing is started in the order in which the processing of a received document is accepted.

The hold queue 19 temporarily holds a document. If a document has the designation of job hold, the job is queued by this hold queue 19 regardless of whether the job is of completion or noncompletion type. The document queued in the hold queue 19 is released from the queued state by the user's instruction. If the job is a completion type job, the document is moved to the spool queue 18, whereas if the job is a noncompletion type job, the document is moved to the printer queue 22.

The wait queue 20 is one type of processing wait queue. When the job which is waiting for a user's password input timed out, the job is moved from the printer queue 22 to this wait queue 20. If a password is input by the user, the job returns to the printer queue 22.

The pause queue 21 holds queues subjected to temporal interruption of processing. If the system manager performs predetermined operation through a server manager (not shown), a job will move from the printer queue, which will be described later, to this pause queue 21, and vice versa.

The printer queue 22 (22-1 to 22-N) is provided corresponding to job output sections (not shown) of the job execution section 13. Jobs waiting for print processing are queued in this queue.

Jobs whose all documents were subjected to processing completion are queued in the terminate queue 23 (23-1 to 23-N).

The job execution section 13 is made up of more than one job processing sections or job output sections (neither being shown in the drawings). The job processing section converts a print format of a job, and the job output section outputs a job, that is, prints a job. The job output section and the printer queue (22-1 to 22-N) are provided in a one-to-one correspondence with each other. A job is queued in the printer queue corresponding to a selected job output section. Further, upon completion of one job, the job execution section 13 issues an acknowledgement of the completion of the processing to the job scheduling device 12. When the job execution section 13 becomes enabled, the job executions section 13 issues an acknowledgment of enabled processing to the job scheduling device 12.

ceeds to step S1310, and the next document will be processed. If the final document identifier is set to TRUE in step S1314, $N = N + 1$ (step S1315), the processing proceeds to step S1304, and the next job will be processed.

In this processing, it is checked whether or not the processing start wait is set to a job, before it is checked whether or not the leading document of the job can issue a processing request. If the processing start wait is set to the job, the job is rendered in a processing start wait state. No processing request is issued to a document of that job and documents of subsequent jobs until the job is released from the processing start wait state by the user's instruction or a timeout. As a result of this, it becomes possible to control a job processing start wait in parallel with the continuous issue of document processing requests.

In the job scheduling device 12 of the second embodiment, processing procedures for the continuous issue of document processing requests and the control of a job processing completion wait will now be described using a flowchart shown in Fig. 14. Here, the explanation of overlaps between this flowchart and the flowchart shown in Fig. 12 will be omitted as needed.

To begin with, upon receipt of a document delivered from the terminal 11, the request control section 14 accepts this document (step S1401) and accepts an acknowledgement of processing completion from the job execution section 13 (step S1402).

The flow of the acceptance of the processing completion acknowledgement issued from the job execution section 13 which is executed in step S1402 will be described using a flowchart shown in Fig. 15.

Upon receipt of an acknowledgement of processing completion from the job execution section 13 (step S1402-1), the job scheduling section 15 acquires document information of a document whose processing was completed (step S1402-2), and the document is placed in a processing completed state (step S1402-3). Subsequently, it is judged whether or not the final document identifier of that document information is TRUE (step S1402-4). If the final document identifier is not TRUE, the processing returns to step S1402-1. If the final document identifier is TRUE, that is, if the document whose processing was completed is the final document, the job information of that document is obtained (step S1402-5). Then, it is judged whether or not a processing completion wait is set to the job information (step S1402-6). If the processing completion wait is set to the job information, the job is rendered in a processing completion wait state (step S1402-7). On the other hand, the processing completion wait is not set to the job information, the processing of that job is completed (step S1402-3).

In the flowchart shown in Fig. 14, it is judged whether or not the printer queue includes a job waiting for processing completion (step S1403). If the queue does not include any job waiting for processing completion, N representing a printer queue number is set to

one (step S1404), and it is judged whether or not the job execution section 13 becomes enabled (step S1405). If the job execution section 13 is enabled, and if a job identifier (a job ID) is obtainable from the job information of an N th job in the printer queue (step S1406), it is judged whether or not a job identifier of the job waiting for processing is set to the printer queue (step S1407). If the job identifier of the job waiting for processing is set to the printer queue, or if there is a job waiting for the completion of processing in step S1403, the processing returns to step S1401 when the job is released from the processing completion wait state by the user's instruction delivered from the terminal 11 or a timeout (step S1408).

If the job identifier of the job waiting for processing is not set to the printer queue in step S1407, it is judged whether or not a processing completion wait is set to the job information of that job (step S1409). If the job completion wait is set to the job information, the job is rendered in a processing completion wait state, and the job identifier of this job waiting for the completion of processing is set to the printer queue (step S1410).

On the other hand, if the processing completion wait was not set to the job information of the job in step S1409, or if the job identifier of the job waiting for the completion of processing was set to the printer queue in step S1410, M representing a document number of that job is set to one (step S1411), and it is judged whether or not M th document information of a job having the previously mentioned job identifier is obtainable (step S1412). If the M th document information is not obtainable, the processing will return to step S1401. However, if the M th document information is obtainable, it is judged whether or not the document information is in a state of requesting processing (step S1413). If it is not in that state, it is then judged whether or not the document information is waiting for processing (step S1414). If the document information is waiting for processing, a processing request for that document is issued to the job execution section 13, and the document information is placed in the state of requesting processing (step S1415).

If the document information is the state of requesting processing in step S1413, or if a document processing request was issued in step S1415, it is judged whether or not the final document identifier of the document information is set to TRUE (step S1416). If the document identifier is not set to TRUE, $M = M + 1$ (step S1417), the processing proceeds to step S1412, and the next document will be processed. If the final document identifier is set to TRUE in step S1416, $N = N + 1$ (step S1418). The processing proceeds to step S1405, and the next job will be processed.

In this processing, it is checked whether or not the processing completion wait is set to a job, before it is checked whether or not the leading document of the job can issue a processing request. If the processing completion wait is set to the job, a job identifier of that job is set to the printer queue. Hereafter, a processing request

ing for the completion of processing are queued, and a terminate queue.

In Fig. 17, the jobs 2 - 5 are queued in the printer queue 22, and the job 1 the processing of which was completed with respect to all documents of that job is queued in the terminate queue 23. In the case of jobs 2 and 3 in the printer queue 22, processing requests have already been issued to all of the documents of these jobs, but the processing of some of the documents is not completed yet, and hence the documents still remain in the printer queue 22.

In this example, a processing completion wait is specified for the job 4, and a job identifier of this job 4 is retained in the printer queue 22. In this case, a processing request is only issuable with respect to documents of the job 4, and no processing request is issued to the next job 5.

When the processing of all the documents of the jobs 2 and 3 are completed, these two jobs move to the terminate queue 23 as shown in Fig. 18. The job 4 is released from the processing start wait state the moment at which the processing of all the documents of the job 4 is completed. Since the job 4 is in the processing completion wait state, a processing request for the job 5 is not issued. After having been released from the processing completion wait state by the user's instruction or a timeout, the job 4 moves to the terminate queue 23.

Fig. 20 is an explanatory view showing the relationship between a printer queue, in which jobs waiting for a password input are queued, a terminate queue, and a wait queue.

Jobs 4, 6, 8, and 9 are queued in a printer queue 22-1, and the job 1 the processing of which was completed is queued in a corresponding terminate queue 23-1. Jobs 3, 7, 10, and 11 are queued in a printer queue 22-2, and a job 2 the processing of which was completed is queued in a corresponding terminate queue 23-2. In the case of jobs 4 and 6 in the printer queue 22-1, processing requests have already been issued to all of the documents of these jobs, but the processing of some of the documents is not completed yet, and hence the documents still remain in the printer queue 22-1.

In the printer queue 22-1, a password input wait is specified for the job 8, and the job 8 enters a password input wait state the moment at which a processing request for the document 2 of the job 6 is issued and the job 8 becomes the next object of the processing. When a password is input by the user, the job 8 is released from the password input wait state. However, if a password is not input during a fixed period of time, the job 8 moves to the wait queue 20 as a result of the occurrence of a timeout. During this period, a processing request for a document of the job 9 is not issued. After the processing of the documents of the jobs 4 and 6 has been completed, these two jobs move to the terminate queue 23-1. At this time, if the job 8 is released from the password input wait state, the processing of the job 8

becomes possible. However, if the job 8 remains in the password input wait state, processing requests for the documents of the jobs 8 and 9 are not issued until the job 8 is released from the password input state. Thereafter, if the job 8 moves to the wait queue 20 as a result of the occurrence of a timeout, a processing request for the job 9 becomes issuable.

The printer queue 22-2 shows the manner in which the job 5 in the password input wait state between the jobs 3 and 7 moved to the wait queue 20 as a result of the occurrence of a timeout. In the printer queue 22-2, the job 7 becomes issuable a processing request, because the job 5 moved to the wait queue 20. The job 5 moved to the wait queue 20 is released from the password input wait state resulting from the input of a password by the user, and the job 5 returns to the printer queue 22-2. When the job 5 returns to the printer queue 22-2, the job is queued at the head of the queue or a priority position thereafter.

As previously described, in the second embodiment, when the job execution section becomes enabled, processing requests are sequentially issued for documents if the processing requests are issuable for these documents. If a processing start wait is set to the leading document of a job, the job is rendered in a processing start wait state, and no processing request is issued to documents of that job and subsequent jobs until the job is released from the processing start wait state by the user's instruction or a timeout. By virtue of this configuration, it is possible to control the processing start wait of a job while processing requests for documents are continuously issued.

Secondly, when the job execution section becomes enabled, processing requests for documents are sequentially issued if processing requests are issuable with respect to the documents. If a processing completion wait is set to the leading document of a job, the job is rendered in a processing completion wait state when the processing of that job is completely finished, and no processing request is issued to documents of subsequent jobs until the job is released from the processing completion wait state by the user's instruction or a timeout. By virtue of this configuration, it is possible to control the processing completion wait of a job while processing requests for documents are continuously issued.

Thirdly, when the job execution section becomes enabled, processing requests for documents are sequentially issued if processing requests are issuable with respect to the documents. If a password input wait is set to the leading document of a job, the job is rendered in a password input wait state when the processing of that job is completely finished, and no processing request is issued to documents of subsequent jobs until the job is released from the password input wait state by the user's instruction or a timeout. By virtue of this configuration, it is possible to control the password input wait of a job while processing requests for documents are continuously issued.

The job execution section control section 108 transfers a processing request from the request control section 106 or the job scheduling section 107 to the job execution section 104. Further, the job execution section control section 108 transfers a status acknowledgement relating to the job delivered from the job execution section 104 to the job scheduling section 107. For example, when a conversion processing section, which will be described later, becomes enabled, the job execution section control section 108 issues a ready acknowledgement to the job scheduling section 107. Contrary to this, if the conversion processing section is disabled, the job execution section control section 108 issues a busy acknowledgement to the job scheduling section 107. A status acknowledgement other than a job is transferred to the object management section 105, and the state of an object which is the object of the status acknowledgement is changed.

The queue management section 109 prepares queues, each corresponding to each job execution section 104, according to an instruction from the job scheduling section 107 and queues jobs. In addition to the preparation of queues, the queue management section 109 moves queues and switches the order of documents in the queue in response to a request from the system manager.

Fig. 22 is a block diagram showing the detailed configuration of the job scheduling section 107.

The job scheduling section 107 is provided with a print document type determination unit 111, an output device selection unit 112, a conversion processing section selection unit 113, and a job search unit 114. In addition to the previously mentioned scheduling function, the job scheduling section 107 searches for each queue in the queue management section 109 during a print processing wait, and carries out the conversion of a print format if there is a job which needs the conversion of a print format.

The print document type determination unit 111 determines whether the processing of a job is a completion type or a noncompletion type with reference to a job request transferred from the request control section 106. The completion type job is characterized in that print processing is not started until all documents included in a job are completely obtained. For example, a one-job multidocument type job is mentioned as one example of this type. The noncompletion type job is characterized in that the processing of a received job is sequentially started.

The output device selection unit 112 selects either of output devices 116-1 and 116-2 depending on an output device name specified for the job, for example, a printer name.

The conversion processing section selection unit 113 judges whether or not the conversion of a print format is necessary by comparing formats of documents included in a job with print formats which the output devices 116-1 and 116-2 can interpret. If the conversion of a print format is necessary, the conversion process-

ing section selection unit 113 selects a conversion processing section suitable for that conversion from conversion sections 115-1 and 115-2.

The job search unit 114 searches spool queues and printer queues using a queue management section 109 which will be described later, and judges whether or not there is a job including documents which need the conversion of a print format at the job execution section 104.

The job execution section 104 is made up of a conversion processing section (a converter) for converting a print format, and an output device (a printer) which outputs a document on a printer in compliance with a print format. The conversion section 115-1 converts a print format A to a print format B, whereas the conversion section 115-2 converts a print format C to a print format D. The output device 116-1 corresponds to the print format B, and the output device 116-2 corresponds to the print format D.

The queue management section 109 prepares various types of queue which the job scheduling section 107 uses. Documents of a completion type job are queued in a spool queue 117, and the job is managed within this queue until all documents of that job are completely obtained. A wait queue 118 is one type of processing wait queue, and jobs waiting for a password input by the user are stored in this queue. A pause queue 119 is a queue in which temporarily interrupted jobs are stored. When a pause button (not shown) of the server management section 102 is pressed, a job moves from a printer queue, which will be described later, to the pause queue 119, or the job returns to the printer queue when a resume button is pressed. Printer queues 121 and 122 correspond to the output devices 116-1 and 116-2, jobs waiting for print processing are queued in these printer queues.

The operation of each section associated with the flow of processing, from the receipt of a document for which a job requested is issued to the queuing of the job in a printer queue, in the previously mentioned print processing system 110 will be described with reference to a flowchart of Fig. 24 and Figs. 22 and 23.

To begin with, the job acceptance section 101 accepts a job request from a terminal connected to a network (not shown) and transfers the thus accepted job request to the job control section 103.

Upon receipt of the job request (step S2101), the request control section 106 of the job control section 103 prepares an object associated with the job request using the object management section 105, and issues a request for job processing to the job scheduling section 107 or the job execution section control section 108. The object prepared by the object management section 105 is used later when conversion or output processing is carried out in the job execution section 104.

When the job request is transferred to the job scheduling section 107, the print document type determination unit 111 determines whether the processing of the job is a completion type job or a noncompletion type

conversion processing from the object management section 105, and converts a print format C included in the job into a print format D. At this time, the conversion section 115-2 issues a busy acknowledgement to the job execution section control section 108 when conversion processing is started, and issues a ready acknowledgement to the job execution section control section 108 when the conversion is completed.

If the selected conversion section 115-2 is not enabled, the job scheduling section 107 leaves the received document queued in the printer queue 122.

If the received document is not the final document of the job (step S2121), the processing returns to step S2101 and the next document is accepted. If the received document is the final document of the job, the processing is terminated.

If the job is a noncompletion type job, documents are sequentially queued in the printer queue 122 from the first document. Depending on the state of the conversion section 115-2 at that time, some documents may be queued in the printer queue 122 while they remain unconverted. However, an attempt is again made to convert these unconverted documents while the documents are queued in the printer queue 122. This processing will be described with reference to the flowchart shown in Fig. 25.

Upon receipt of a ready acknowledgement from the conversion processing section which was not possible to process the job in step S2109 or step S2119 shown in Fig. 24 via the job execution section control section 108, the job scheduling section 107 retrieves a job including a document which needs conversion and makes again an attempt to convert the document. This processing is carried out in parallel with the previously mentioned processing defined by the flowchart shown in Fig. 24.

The flow of conversion processing when the conversion processing section becomes enabled will be described using a flowchart shown in Fig. 25, and the operation of each section will also be described with reference to Figs. 22 and 23.

Upon receipt of a ready acknowledgement from the conversion section 115-1 or 115-2 via the job execution section control section 108 (step S2201), the job scheduling section 107 searches each queue managed by the queue management section 109 using the job search section 114 and retrieves a job including a document which needs conversion processing carried out in the conversion processing section.

Initially, the job search unit 114 sets a count N to one (step S2202), and retrieves a job, including a document which need the conversion of a print format executed by the conversion processing section, from a printer queue N (step S2203).

In other words, jobs stored in a queue to be retrieved are sequentially searched, and it is judged whether or not the searched job includes a document which needs the conversion of a print format. For example, when the output device 116-1 is selected with

respect to the searched job, it is judged whether or not this job includes a document in a format other than a print format B, because the output device 116-1 can process the print format B. If the output device 116-2 is selected with respect to the searched job, it is judged whether or not the job includes a document in a format other than a print format D, because the output device 116-2 can process a print format D.

If the job including a document which needs conversion is found, a conversion processing section which carries out necessary conversion is selected, and a request for the conversion of the document is issued to the selected conversion processing section via the job execution section control section 108 (step S2204). The conversion processing section reads information necessary for conversion processing from the object management section 105 and converts a print format included in the job. The conversion processing section issues a busy acknowledgement to the job execution section control section 108 when the conversion processing is carried out and also issues a ready acknowledgement to the job execution section control section 108 when the conversion is terminated.

If the job including a document which needs conversion is not found in the printer queue N, it is judged whether or not there is another printer queue (step S2205). If there is another printer queue, N+1 is reset to N (step S2206), and the processing returns to step S2203. If there is no printer queue to be searched in step S2205, the pause queue 119 will be searched to find a job including a document which needs the conversion of a print format executed in the conversion processing section (step S2207). If a job including a document which needs conversion is found, a conversion processing section which carries out necessary conversion processing is selected, and a request for the conversion of the document is issued to the selected conversion processing section via the job execution section control section 108 (step S2204).

If a job including a document which needs conversion is not included in the pause queue 119, the wait queue 118 will be searched to find a job including a document which needs the conversion of a print format executed in the conversion processing section (step S2208). If a job including a document which needs conversion is found, a request for the conversion of the document is issued to a conversion processing section via the job execution section control section 108 (step S2204).

If a job including a document which needs conversion is not included in the wait queue 118, the spool queue 117 will be searched to find a job including a document which needs the conversion of a print format executed in the conversion processing section (step S2209). If a job including a document which needs conversion is found, a request for the conversion of the document is issued to a conversion processing section via the job execution section control section 108 (step S2204).

work N, and it is also possible for the client to issue a request for the acquisition of necessary information, for example, printing completion to the management I/O control section.

The job control section 203 receives a print processing job from the job acceptance section 201 and analyzes the contents of the job. The job control section issues a request for the execution of that job to the job execution section 204. The job control section 203 also receives a control request for management from the management section 202 through the management I/O control section 202a and executes processing corresponding to the contents of the request.

The job execution section 204 is a printer to execute a job such as printing or a converter which executes a job for converting the format of the print document, with respect to which the client issued a print request, to a page description language supported by the printer, for example, PostScript. Accordingly, an execution instruction is not necessarily issued once to the job execution section 204 from the job control section 203 depending on the contents of the job. The execution instruction is repeatedly issued a necessary number of times. A device which executes the job may be a various types of printer or converter. In the illustrated embodiment, as one example, three job execution sections 204a, 204b, and 204c are shown.

The event processing section 210a issues an acknowledgement to the management section 202 via the management I/O control section 202a with respect to various types of event such as the completion of a job and the cancellation of a job issued from the job control section 203. The history of issue of various event is written into the log file 206 through the logging processing section 210b. The accounting processing section 210c collects setting formation relating to jobs such as what kind of function was utilized. Account information relating to, for example, the number of pages of a printed document requested through the object management section 210d and the object processing section 208 is collected from the object file 209, and the thus collected information is written into the account file 207.

The object management section 210d manages jobs accepted by the job acceptance section 201 as one object. The object management section 210d writes attribute information relating to a job requested by the client into the object file 209 using the object processing section 208, and reads information of a necessary object from the object file 209 through the object processing section 208 as needed. When the job execution section 204, which is a printer, fetches document data of a print document from the spool of the job acceptance section 201 to print the document, the object processing section 208 also reads document attribute, such as the location of document data and paper size on which the data are printed, from the object file 209 and sends that document attribute to the job execution section 204.

Fig. 28 is a block diagram showing one example of the configuration of the job control elementary section 210.

In Fig. 28, the job control elementary section 210 is further provided with a request control section 211, a job scheduling section 212, a job execution section control section 213, and a queuing management section 214. The request control section 211 executes processing between the request control section 211 and the job acceptance section 201, between the request control section 211 and the management section 202 through the management I/O control section 202a, and between the request control section 211 and the object management section 210d. The job execution section control section 213 executes processing between the job execution section control section 213 and the job execution section 204.

The request control section 211 accepts a job request from the job acceptance section 201 and analyzes the job request. If the job request is recognized as job information, the request control section 211 issues a request for the generation of job information to the object management section 210d. Based on the thus generated information, the request control section 211 issues a request for scheduling to the job scheduling section 212. Further, the request control section 211 transfers a request from the management section 202 which is irrelevant to the job, for example, the interruption of a printer, to the job execution section control section 213, and sends back response from the job execution section 204 which is received from the job execution section control section 213 to the management section 202.

The job scheduling section 212 fetches a corresponding job and an object of the associated job execution section 204 from the object management section 210d. The job scheduling section 212 schedules the job, and carries out the queuing of the job, such as the preparation of a specialized queue corresponding to each of the job execution sections 204, using the queue management section 214. The job scheduling section 212 issues a job processing request to the job execution section 204 through the job execution section control section 213. The queue management section 214 queues the job in compliance with an instruction from the job scheduling section 212, and sends information necessary for accounting processing to the accounting processing section 210c.

Corresponding to the acceptance of the jobs, the request control section 211 generates a plurality of request processing sections 220-1, 220-2, ... 220-n. Similarly, the job scheduling section 212 generates a plurality of processing sections 230-1, 230-2, ... 230-n which carry out processing in response to a request from the request control section 211 and a plurality of different processing sections 240-1, 240-2, ... 240-n which carry out processing with respect to the job execution section control section 213. The processing is efficiently carried out in a parallel manner.

the description of a document "document-description";
 the designation of a page description language "document-format";
 the designation of an output bin "output-bin-name";
 the designation of paper size "media-used";
 the designation of a paper feeder "feeder-used";
 and
 the designation of single/double-sided printing "sides".

These attribute elements can be specified for each document.

For the data 280 sent from the client, since there a plurality of document data items in the data portion of the data 280, it can be understood that print requests for a plurality of documents must be processed as one job.

The processing of the job scheduling section 212 when a print request was properly accepted from the client without a failure will be described.

Fig. 31 is a flowchart showing the flow of the processing of the job scheduling section.

This flowchart shows processing carried out when a print request is normally received without any failures during timeout monitoring practiced by the time monitoring section 253 between received documents.

When the request control section 211 accepted a job, it is checked whether or not the attributes, i.e., parameters of the job and a document of that job are valid before the job is processed by the job scheduling section 212 (step S1). Subsequently, it is checked whether or not the check is valid (step S2). If it is judged that the job includes invalid parameters, the processing will proceed to error processing (step S3).

The schedule attribute judgement section 251 judges whether the scheduling attribute is the after-complete processing or the before-complete processing (step S4). If the scheduling attribute is judged as the after-complete processing, the job scheduling section 212 queues the received document in the reception-waiting queue 270 of the queue management section 214 (step S5). Then, the final document judgement section 252 judges whether or not the received document is the final document and all documents have been received (step S6). If all documents were not received yet, the final document judgement section waits for the reception of the next document. If the reception of the document has been completed, the processing proceeds to the selection of a printer by the printer selection processing section 256 (step S7). The document is moved from the reception-waiting queue 270 to the printer queue 260 (step S8).

If the scheduling attribute is judged as the before-complete processing as a result of the judgement in step S4, it is checked whether or not the received document is the leading document (step S9). If the received document is the leading document, the printer selection processing section 256 selects a printer (step S10). If the received document is not the leading document,

printer selection is omitted because the printer has already been selected, and the document is queued in the printer queue 260 (step S11).

It is then checked whether or not the queuing of the document has properly been performed (step S12), and it is also checked whether or not the selected printer is ready for printing (step S13). If the printer is read for printing, a print request is issued to the printer by way of the job execution section control section 213 (step S14). On the other hand, if the printer is occupied and disabled, the processing enters a standby state until the printer becomes enabled.

Fig. 32 is a flowchart showing the flow of timeout processing in the case of the after-complete processing.

According to this flowchart, if a receiving failure arises during timeout monitoring practiced between received documents, it is checked whether or not the accepted job has the attribute of after-complete processing. The final document judgement section 252 judges whether or not a document received before the occurrence of the receiving failure is the final document, that is, whether or not all of the documents have already been received (step S21). If all of the documents have already been received, the processing proceeds to the selection of a printer. If it is judged that some documents are not received yet in step S21, the final document processing section 255 sets the finally received document queued in the reception-waiting queue 270 to the final document of this job, namely, the final document processing section sets final document information with respect to the finally received document (step S22), and the processing proceeds to the selection of a printer.

Fig. 33 is a flowchart showing the flow of timeout processing at the time of before-complete processing.

If a receiving failure arises during timeout monitoring practiced between received documents, the schedule attribute judgement section 251 checks that the received job has the attribute of before-complete processing. To begin with, it is checked whether or not the reception of the job has been completed (step S31). If it is checked that the reception of the job has already been completed, the job will be completed. However, if the reception of the job has not been completed yet, it will be checked whether or not the printing of all documents received by that time has been completed (step S32). If the printing of all the documents has already been completed, the job will be completed. Hence, the job completion processing section 254 carries out the completion processing of the job, that is, removes the leading job block from the printer queue 260 (step S33). In step S32, if it is judged that the printing of all the documents is not completed yet and some documents are waiting for printing, the job completion processing section 254 sets the final document which is queued in the printer queue 260 and is waiting for printing as the final document of this job (step S34), and the processing continues.

acceptance section 301 or the management I/O control section 302a, and prepares a job object using the object management section 310d. The request control section 311 also issues a request for processing to the job scheduling section 312. The job execution section control section 313 transfers a processing request from the request control section 311 or the job scheduling section 312 to the job execution section 304.

When received a job request via the request/acknowledgement acceptance section 312a, the job scheduling section 312 executes the processing of a command by activating a command processing section 3012b corresponding to a command included in the job request. The job scheduling section 312 is a processing section which issues a job execution instruction to the job execution section control section 313 via a transmission processing section 312c, and also executes the queuing of a job using a queue management section 312d. It is possible to make such a job request include a pause command and a resumption command. When the pause command is received, a command processing section (hereinafter referred to as a pausing section 312e) for the pause command is activated. On the other hand, when the resumption command is received, a command processing section (hereinafter referred to as a resuming section 312f) for the resumption command is activated.

The queue management section 312d manages a printer queue 312g and a pause queue 312h. The printer queue 312g is provided corresponding to a printer so as to queue job information of jobs waiting for printing. A job assigned to each printer is queued in a corresponding printer queue. In other words, the printer queue 312g is provided corresponding to physical printers included in the job execution section 313. A pause queue 312h queues jobs paused according to a user's instruction. Jobs held in the pause queue 312h are retained in this queue until a resumption instruction is issued by the user. When the user issues the resumption instruction, the job information will be moved to the assigned printer.

For convenience of explanation, only the printer queue 312g and the pause queue 312h are illustrated in this embodiment. However, it is also possible to use a spool queue for retaining a job until all documents included in that job are accepted, a hold queue for queuing unscheduled jobs which are not subjected to scheduling, a wait queue for queuing jobs which timed out as a result of waiting for a password input, and a terminate queue provided corresponding to printers in order to queue only printed jobs.

The configuration and processing procedures of the pausing section 312e and the resuming section 312f will be described more detail with reference to Figs. 36 to 39.

Fig. 36 shows the configuration of the pausing section 312e shown in Fig. 34. As shown in Fig. 36, the pausing section 312e is made up of a job state control section 330a, a queue state control section 330b, a

printing job control section 330c, and a termination processing section 330d. The pausing section 312e temporarily interrupts a job which is being printed and a job waiting for printing.

The job state control section 330a controls the state of a job and the entire pausing section 312e. Specifically, when a pause command is received, it is checked whether or not the specified job is being printed or waits for printing. Only when the job is being printed, the job state control section 330a issues a printing interruption instruction to the printing job control section 330c. Upon receipt of a job interruption acknowledgement from the job execution section 304, the job state control section 330a carries out the modification of a job status, the issue of an queue status modification instruction to the queue state control section 330b, and the issue of a completion instruction to the completion processing section 330d. When the modification of the states of a job and a queue has been completed, the job state control section 330a prepares recovery information and affords the thus prepared recovery information to the completion processing section 330d. This recovery information includes the check points which the job execution section 304 wrote into the object file 309. The queue state control section 330b carries out the modification of the state of the queue associated with the pausing of the job. Specifically, the queue state control section 330b moves the job from the printer queue 312g to the pause queue 312h in response to the modification instruction from the job state control section 330a.

Upon receipt of a job interruption instruction from the job state control section 330a, the print job control section 330c issues an instruction for the interruption of job processing to the job execution section 304 via the transmission processing section 312c. When issuing an interruption instruction to the job execution section 4, the printing job control section 330c issues that instruction based on conditions of the issue of jobs. Specifically, when a one-job multidocument is processed, print processing requests for a plurality of documents are issued to the job execution section 304. The order of the issue of these requests is retained in a job issue management table 331, and the documents which issued the print processing requests later are subjected to an interruption instruction prior to other documents. This is intended to avoid the situations in which the documents to which the interruption instruction was issued have already been printed as much as possible, and it is also intended to prevent a subsequent document from proceeding to the next processing phase and from being printed before the next interruption instruction is notified.

The termination processing section 330d notifies a workstation, which issued the pause request, of a processing result showing whether or not the specified job paused.

The use of the pausing section 312e having the previously mentioned configuration allows each job to pause.

A specific example of the states of the printer queue 312g and the pause queue 312h associated with the previously mentioned pausing and resuming processing will be described.

Figs. 40a to 40c show one example of the states of the printer queue 312g and the pause queue 312h when pausing and resuming processing are carried out. In this illustrative embodiment, an explanation will be given of the pause and resumption of a job A which is being processed based on the assumption that jobs A, B, and C are originally included in the printer queue 312g.

As shown in Fig. 40a, on the assumption that the printer queue 312g retains the job A (which is being processed) and pending jobs B and C, when the pause command is issued with respect to the job A, the job A is passed to the pause queue 312h as shown in Fig. 40b, and the printer queue 312g starts to process the next job B. When a resumption command is issued in this state, the pausing job A retained in the pause queue 312h is queued to the end of the printer queue 312g, as shown in Fig. 40c. In this way, the pausing job is moved between the printer queue 312g and the pause queue 312h, whereby it becomes possible to cause each job to pause or resume.

A processing sequence mutually executed between the job scheduling section 312 and the job execution section 304 will be described.

Fig. 41 is a schematic representation showing a processing sequence executed between the job scheduling section 312 and the job execution section 304. Hereinbelow, an explanation will be given of the processing of a one-job multidocument in which a job consists of a document 1 (hereinafter referred to as Doc 1) and a document 2 (hereinafter referred to as Doc 2).

As shown in Fig. 41, when the job scheduling section 312 issues a request for printing Doc 1 to the job execution section 304 (step S801), the job execution section 304 sends back an acceptance acknowledgement, representing the acceptance of the request together with a ready acknowledgement, representing that it can accept the print processing request, to the job scheduling section 312 (step S802). The printing preprocessing of Doc 1 is carried out (step S803), and the printing of Doc 1 is then started after the preprocessing has been completed (step S804).

Thereafter, when the job scheduling section 312 issues a request for printing Doc 2 to the job execution section (step S805), the job execution section 304 sends back the acceptance acknowledgement together with a busy acknowledgement which represents that it cannot accept a print request to the job scheduling section 312 (step S806). The printing preprocessing of Doc 2 is started (step S807).

If the job scheduling section 312 accepted the pause command, the job scheduling section recognizes that it should issue a printing interruption request for the documents in the order of Doc 2 and Doc 1 upon reference to the job issue management table 331.

When the job scheduling section 312 issues the printing interruption request for Doc 2 toward the job execution section 304 (step S808), the job execution section 304 sends the acceptance acknowledgement back to the job scheduling section 312 (step S809) and interrupts the printing preprocessing of Doc 2 (step S810).

The job scheduling section 312 which received the acceptance acknowledgement issues a printing interruption request for Doc 1 toward the job execution section 304 (step S811). In response to this, the job execution section 304 sends an acceptance acknowledgement back to the job scheduling section 312 (step S812), and terminates the printing of Doc 1.

Upon recognition of the success of the interruption of Doc 2 (step S814), the job execution section 304 notifies the job scheduling section 312 of the success of the interruption of Doc 2 (step S815). Further, upon recognition of the success of the interruption of Doc 1 (step S816), the job execution section 304 notifies the job scheduling section 312 of the success of the interruption of Doc 1 (step S817).

Through the above mentioned processing sequence, the job scheduling sequence 312 can confirm that Doc 1 and Doc 2 included in the job paused.

As has been described above, in this embodiment, the pausing section 312e moves a job stored in the printer queue 312g from the printer queue 312g to the pause queue 312h in response to the pause command for that job. Further, the resuming section 312h moves the job stored in the pause queue 312h to the printer queue 312g in response to a resumption command for that job. The job execution section 304 sequentially executes jobs stored in the printer queue 312g. Such a configuration results in the following advantages:

- 1) It becomes possible to easily change the attribute of a job retained in the printer queue 312g.
- 2) It becomes possible to improve the processing efficiency of a job through interruption and resumption executed on a job-by-job basis.

Furthermore, according to this embodiment, the provision of a new check point as a parameter for a resumption command allows the resumption of the processing of an arbitrary document by removing the processing of a specific document.

In this embodiment, the job execution section 304 received a pause instruction is arranged so as to store a document number, a page number and a designated number of copies as check points into the object file 309. However, the present invention is not limited to such a configuration, and information necessary to resume an availability of collation can be stored as a check point in the object file.

Moreover, although the resuming section 312f queues a job to the end of the printer queue 312g when processing a resumption command in this embodiment,

mation retained in this pause queue q4 is retained in this queue until the user instructs the resumption of the job. When the user instructs the resumption of the job, the job information is moved to an assigned printer queue.

The printer queues q5 and q6 are provided corresponding to printers so as to queue job information of jobs which are waiting for printing. Only job information of a job assigned to each printer is queued in a corresponding printer queue.

In this embodiment, the printer queue 5 corresponds to either one of two printers provided in the job execution section 413, and the printer queue q6 corresponds to the other printer.

A processing request is issued with respect to the job information retained in the printer queue q5 or q6 if the printer becomes enabled and if the leading job in the printer queue q5 or q6 includes a processable document. When the print processing of the printer is completed, the job information of that job is moved to the terminate queue q7 or q8.

The terminate queues q7 and q8 are provided corresponding to the printers in order to queue job information of printed jobs. Only job information of a job assigned to each printer is queued in a corresponding terminate queue.

As a result of the use of the job scheduling device 412 having the above configuration, it becomes possible to manage a queue corresponding to the state of a job and hence to effect appropriate scheduling.

In this embodiment, an explanation will be given of the case where the job execution section 413 has two printers. The number of printer queues and terminate queues is determined corresponding to the number of registered printers retained in the job execution section 413.

The processing procedures of the scheduling device 412 will now be described. Here, the processing of the scheduling device, which recovery is required in response to the interruption of processing, will be described later.

Fig. 43 is a flowchart showing processing procedures of the scheduling device 412 shown in Fig. 42 between the receipt of a job and the execution of print processing.

As shown in Fig. 43, upon receipt of a job output from the terminal 411a or 411b via the network 410 (step S4201), the request control section 412a checks whether or not a job ID, which will be described later, is set to the job (step S4202). If the job ID is set to the job, job information is obtained from this job ID (step S4203), and then it is checked whether or not the accepted document is the final document (step S4204).

If the document is not the final document, the job ID is sent back to the terminal which issued the request (step S4205). On the other hand, if the document is the final document, the number of documents is written into the job ID (step S4206), and the job ID is sent back to the terminal which issued the request (step S4205).

Contrary to this, if the job ID is not set to the job, a job ID is newly set to the job and job information is prepared (step S4207), and the job ID is sent back to the terminal which issued the request (step S4205).

The job scheduling section 412b which accepted the job information of that job checks whether or not a job hold is specified for that job (step S4208). If the job hold is not specified for that job, the job is queued in the hold queue q2 (step S4209).

On the other hand, if the job hold is not specified for that job, or if the user issued a hold cancellation instruction with respect to the job queued in the hold queue q2 (step S4210), it is checked whether or not that job is an AC job (an after-complete job) (step S4211). If the job is the AC job, the job is queued in the spool queue q1 (step S4212).

Contrary to this, if the job is not the AC job, or if all documents included in the job queued in the spool queue q1 are completely accepted (step S4213), a printer which carries out printing is selected, and the job is queued in a printer queue corresponding to the selected printer (step S4214).

When a processing turn of the job queued in the printer queue came, a document included in that job is printed (step S4215).

If the job is a one-job multidocument job, all documents included in that job are printed (steps S4215 and S4216).

After the job has been output to the terminate queue (step S4217), printing is completed.

In this way, the job accepted by the job scheduling device 12 is finally retained in the terminate queue.

Subsequently, the configuration of the job information queued in each queue will be described.

Fig. 44 is an explanatory view showing the configuration of job information queued in each queue.

As shown in Fig. 44, the job information comprises a job ID representing an identification number of a job; docsecNum for use in identifying a one-job multidocument; currentDocNum to which the docsecNum for issuing a print request is set; a status 430a which represents the state of a job; prevQID representing a previously retained queue ID; jobCopyCount representing the number of copies; requestCounter representing how many times a print request for a document is issued to a job execution section; completeCounter representing how many times the processing of a document is completed; abortedCounter representing the number of times a job is aborted during the processing of a document; collateFlag representing whether or not collation is carried out; resumeFlag representing whether or not resumption of the job is carried out; previousStatus representing a previous status; retry representing a retry counter when a job is rejected by the job execution section; complete representing whether or not a document is the final document; convertID representing an ID of a requesting converter; ohPagesComp representing how many pages of a job were output; ohDocPagesComp representing how many pages of a document were out-

processing, the recovery processing is carried out in such a way that the recording of the account has been completed.

As a result of this, it is possible to check whether or not the account is recorded.

(1d) If job processing is interrupted when the job execution section 413 can process a plurality of documents at one time, the number of produced copies may become indefinite.

To prevent this, the job scheduling section 412b decrements a requestCounter, which represents the number of requests for printing a document to the job execution section 413, every time the print request is issued to the job execution section 413. Upon receipt of an acknowledgement of the completion of printing processing from the job execution section 413, the job scheduling section 412b increments a completeCounter which represents the number of times document processing is terminated.

If $\text{jobCopyCount} = \text{requestCounter} + \text{completeCounter}$ does not stand when the recovery processing is effected, the recovery processing is resumed by newly setting $\text{requestCounter} = \text{jobCopyCount} - \text{completeCounter}$.

For example, if $\text{requestCounter} = 2$, $\text{jobCopyCount} = 5$, and $\text{completeCounter} = 1$, a sum between the requestCounter and the completeCounter is smaller than the jobCopyCount. Hence, the processing is continued by newly setting "4" to the requestCounter.

As a result of this, even if the processing is interrupted, it becomes possible to appropriately set the number of times requests for printing a document are issued to the job execution section 413.

The processing carried out by the recovery processing section 412e will be described.

The processing of the recovery processing section 412e falls into two categories: namely, processing which is carried out immediately after an abnormal condition occurs, and processing which is carried out when the job scheduling section reboots. These two types of processing will be explained hereinbelow.

The processing carried out immediately after the occurrence of an abnormal condition will first be explained.

The recovery processing section 412e carries out the processing when

(2a) the job execution section 413 detects an abnormal condition within the system, and a document being processed is aborted;

(2b) the job execution section 413 suddenly terminates the processing as a result of the occurrence of an abnormal condition; and

(2c) a converter processing section of printer and converter processing sections of the job execution section 413 which converts a format is abnormally terminated.

Each of these three cases will be described.

(2a) If a job is being suspended or an interrupt wait is being executed when the job execution section 413 is aborted, the occurrence of an abnormal condition is output to a terminal which issued the request. As a result of this, the user can be notified of the occurrence of an error.

If the final document of a one-job one-document job or a one-job multidocument job is being instructed to print when the aborting processing is received from the job execution section 413, the job is aborted.

However, if a document other than the final document of the one-job multidocument job is being instructed to print at this time, the requestCounter of that job is set to -1. Upon receipt of an acknowledgement of the termination of the job scheduling device 413 from a manager (not shown) which supervises the job scheduling device 412 and the job execution section 413, a requestCounter set to -1 is searched, and a job corresponding to that requestCounter is aborted. At this time, if the format of another document is being converted, a termination request is issued to the conversion processing section which carries out the conversion of a format. When the conversion processing section received that termination request, the job is aborted.

Through the execution of postprocessing associated with the occurrence of such an error, it becomes possible to easily effect recovery processing when an error arises.

(2b) If a job is being suspended or an interrupt wait is being executed when the job execution section 413 is suddenly terminated, the occurrence of an abnormal condition is output to a terminal which issued the request in the same manner as in the case of (2a).

If the final document of a one-job one-document job or a one-job multidocument job is being instructed to print when the aborting processing is received from the job execution section 413, the job being processed is rendered in a pending status. When an enabled acknowledgement is received from the job execution section after the job has been restarted, the printing of the pending job is requested again. As a result of this, it is possible to prevent a job from being lost.

However, if an error arises again while the job once rendered in a waiting state by the job execution section 413 is being processed, the user's instruction is needed to restart the job by providing the job with a start-wait attribute. This is intended to prevent the repetition of a print request and errors.

(2c) If a job is being converted (converting) or waiting for response while requesting conversion (convert_requesting) when the conversion processing section of the job execution section 413 which

Subsequently, a processing job, which was receiving abortexit from the job execution section 413 when abnormally terminated, is aborted. Further, jobs which are being converted or waiting for response while requesting conversion are rendered in a conversion request wait status (convert_pending). When a processing enabled acknowledgement is received from the conversion processing section, a request for converting the jobs is issued.

Moreover, a job, which was in the startwait and passwordwait states when abnormally terminated, is again rendered in the startwait and passwordwait states when an enabled acknowledgement is accepted from the job execution section 413.

If there is no document to which a processing request should be issued, and if the processing of a job has already been completed, the job is moved to the terminate queue and is rendered in a terminated status. Then, the reason is sent to complete_with_error.

In the case of a BC job, documents which are queued at the time of recovery are handled as one job. If there is a document without a master file for storing print data of a job, the document is aborted. (3f) Recovery processing of terminate queues q7 and q8

Jobs, retained in the terminate queues q7 and q8 at the time of recovery, are rendered in a terminated status. Further, as regards jobs aborted as a result of the abnormal termination are rendered, the reason of the jobs is set to cancelled_by_user.

Through the execution of a series of the above mentioned processing, the recovery processing section 412e can recover the jobs retained in each queue.

As previously mentioned, according to the present embodiment, upon receipt of a job processing request from the terminal 411a and 411b, the job scheduling section 412b schedules jobs while moving the job from one queue to another queue depending on the state of that job. Hence, it becomes possible to carefully manage jobs.

If any failure arises during the scheduling of jobs, the recovery processing section 412e recovers the previous state of each of jobs retained in the plurality of queues at the time of recovery from the failure. Even if a system failure arises, the influence of the system failure can be reduced, and job processing can be executed as instructed by the user.

Further, in this embodiment, the status of job information is rewritten after the job information has been moved, and recovery processing is carried out based on the practical queue which retains the job. Hence, it is possible to eliminate inconsistency between the status of job information and the practical queue retaining that job information.

In this embodiment, job information is deleted after a master file has been deleted, and if the master file is

not present in spite of the fact that requestCounter of the job information is more than one, recovery processing to delete the job information is carried out. Hence, it is possible to eliminate inconsistency between the job information and the master file.

Furthermore, in this embodiment, the status of job information is changed after an account has been recorded, and if the job information is in a terminated status, recovery processing is carried out in such a way that the account has already been recorded. Therefore, it is possible to check whether or not the account is recorded.

Moreover, in this embodiment, the job scheduling section decrements a requestCounter, which represents the number of requests for printing a document to the job execution section 413, every time the print request is issued to the job execution section 413. Upon receipt of an acknowledgement of the completion of printing processing from the job execution section, the job scheduling section increments a completeCounter which represents the number of times document processing is terminated. In this way, the requestCounter and the completeCounter are matched with each other. Hence, if processing is interrupted, the number of times (requestCounter) a request for printing a document is issued to the job execution section 413 can be appropriately set.

As has been described above, according to the sixth embodiment, upon receipt of a job processing request from a terminal, the job scheduling unit schedules a job while moving the job from one queue to another queue depending on the state of that job. Even if a failure arises during the scheduling of a job, recovery unit recovers the previous state of each of jobs retained in a plurality of queues at the time of recovery from the failure. Even if a system failure arises, the influence of the system failure can be reduced, and job processing can be executed as instructed by the user.

With reference to Figs. 44 to 51, a job processing system according to a seventh embodiment of the present invention will be described. The job processing system of this embodiment is intended to simply modify the attribute of a job. The overall configuration (shown in Fig. 45) of the job processing system of the seventh embodiment is substantially the same as that of the sixth embodiment (shown in Fig. 42), the same reference numerals are provided to designate corresponding features shown in Fig. 42, and hence the detailed explanation thereof will be omitted here for brevity. Moreover, since the configuration of job information of the seventh embodiment is the same as that of the sixth embodiment (shown in Fig. 44), Fig. 44 used in describing the sixth embodiment is quoted to the extent necessary.

In Fig. 45, the job scheduling section 412b resides in an attribute modification section 512e which modifies attribute information in response to a request for modifying the attribute information of a job.

Upon receipt of the request for modifying the attribute of a job, the attribute modification section 512e

ble (step S4309), and hence the processing is terminated.

If the job is not being processed, it is checked whether or not the job information of that job is retained in the hold queue q2 (step S4305). If the job information is held in the hold queue q2, the attribute information of that job is deemed as being changeable.

If the job information of this job is retained in the printer queue q5 or q6, or the pause queue q4 (step S4306), it is further checked whether or not the job is being processed (step S4307).

Specifically, it is checked whether or not the job has previously been processed by checking whether or not the previous state of the job information (previous_job_state) is in a processing state (processing).

As a result of this, if the job was not previously in the processing state, the attribute information of that job is deemed as being changeable (step S4308). On the other hand, if the job was previously in the processing state, the modification of the attribute information of the job is not admitted (step S4309).

If the job information is present in a queue other than the hold queue q2, the pause queue q4, and the printer queue q5 or q6, the modification of the attribute information of the job is not admitted (step S4309).

In other words, when the job information is present in either the terminate queue q7 or q8, the processing of that job has already been completed. Further, when the job information is present in the spool queue q1, the acceptance of an AC job has not been completed yet. Moreover, when the job information is in the wait queue q4, the job is waiting for a password input, and therefore the attribute information of that job should not be changed.

Through the execution of a series of the previously mentioned processing, it is possible to check whether or not the correction of the attribute information of a job is possible depending on the state of the job when an instruction for the modification of the attribute information of that job is accepted.

Subsequently, an explanation will be given of a check as to whether or not the attribute information which the attribute modification section 512e corrects (2) is appropriate.

If the previously mentioned job is in the state in which the attribute information thereof is changeable, the attribute information can be modified in principle. However, the user attempts to provide erroneous attribute information, the modification of the attribute information should not be admitted.

To prevent such erroneous modification of the attribute information, a check is made for the following three points with respect to the attribute information specified and corrected by the user: namely,

(2a) whether or not an attribute not supported by the job scheduling device 412 is specified;

(2b) whether or not multiple values (Multi_Value) are provided as an attribute value to an item where a single value (Single_Value) should be provided; and

(2c) whether or not the attribute information of a job which the second and later documents possess is specified. For (2c), when a one-job multidocument is used, each document possesses the attribute information of a job. On the assumption that the attribute information the first document possesses must be changed in order to change the attribute information of the job, it is checked whether or not the attribute information which the second and later documents have is specified.

Fig. 47 is a flowchart showing procedures for the confirmation of the attribute information carried out by the attribute modification section 512e.

As shown in Fig. 47, upon receipt of the corrected attribute information specified by the user, the attribute modification section 512e checks whether or not the attribute information includes an attribute which is not supported by the job scheduling device 412 (step S4401). If the attribute information includes the attribute which is not supported by the job scheduling device, an error acknowledgement is issued (step S4402), and the processing is terminated.

On the other hand, if the attribute information only comprises the attribute supported by the job scheduling device 412, it is checked whether or not multiple values (Multi_Value) are provided as an attribute value to an item where a single value (Single_Value) should be provided (step S4403). If the multiple values are provided, an error acknowledgement is issued (step S4402), and the processing is terminated.

To the contrary, if the multiple values are not provided, it is confirmed whether or not the job attribute information of the second and subsequent documents for the one-job multidocument is specified (step S4404), and if the attribute information of the second and subsequent documents is specified, an error acknowledgement is issued (step S4402), and the processing is terminated.

If the attribute information of a job of the first document is specified, it is further checked whether or not a document number (document_sequential_number) is specified.

As a result of this, if a document number is specified, the attribute information of only the specified document is changed (step S4406). If the document number is not specified, the attribute information of all documents included in that job is changed (step S4407). The modification of the attribute information is deemed as being changeable (step S4408), and the processing is terminated.

Through the execution of a series of the previously mentioned processing, it becomes possible to reduce the chance that the user provides erroneous attribute information.

default profiles X, Y, and Z, when an instruction 471b for changing the attribute B using the default profile X is received, the value b3 of the attribute B is fetched from the default profile X, and the attribute information 470a is changed using the attribute value b3.

As a result of this, it is possible to obtain new attribute information 471c with the value of the attribute B being changed to b3.

In the case of the one-job multidocument, the number of documents included in a job differs depending on jobs. For this reason, if an instruction for changing an attribute with the use of the default value is issued by specifying not a document but a job, a default value is set to the attribute information of a job which all documents included in the job possess.

However, if the user knows the number of documents included in one job, it is possible to specify the attribute modification section in such a way that the attribute of the default file X is applied to documents 1 and 2 and the attribute of the default file Y is applied to documents 3 and 4.

Processing procedures executed by the attribute modification section 460a will be described.

Fig. 51 is a flowchart showing procedures for modification of attributes carried out by the attribute modification section 460a shown in Fig. 49.

As shown in Fig. 51, to begin with, upon receipt of an instruction for modification of attribute information, the attribute modification section 460a checks whether or not the user specified an attribute value (step S4801). If the attribute value is specified, the attribute value is utilized (step S4802). If the attribute value is not specified, an attribute value retained in the default profile is utilized (step S4803).

If there are a plurality of default profiles, and if a default profile is specified, an attribute value of that default profile is used. However, if the default profile is not specified, the attribute value of the first profile is utilized.

It is checked whether or not the job is the one-job multidocument (step S4804). If the job is the one-job multidocument, the attribute information is corrected using the attribute value (step S4805).

On the other hand, if the job is the one-job multidocument, it is checked whether or not a profile having an attribute to be changed is specified in such a way as to correspond to a document (step S4806). If the profile is not specified corresponding to a document, the attribute information of all documents is corrected using the same attribute value (step S4807).

On the other hand, if the profile is specified corresponding to a document, the attribute information is corrected using an attribute value specified to each document (step S4808), and the processing is terminated.

Through a series of the above mentioned processing, it is possible to change attribute information not only using the attribute value specified by the user but also using the attribute value stored in the default profiles.

As previously mentioned, according to the seventh embodiment, if a print job processing request is received from the terminal 411a or 411b, the job scheduling device 412b schedules print jobs while moving a print job from one queue to another queue depending on the state of the print job. If an instruction for modification of attribute information of the print job is received during the scheduling of the print job, the attribute modification section 512e changes the attribute information only when the print job is in the state wherein the attribute information thereof is changeable, and when the instruction for modification of the attribute information is free from errors. Even if attribute information is erroneously specified, it is possible to easily change the attribute information while reducing a user's burden, a drop in the use efficiency of a network, and print waiting time.

In the modified example, the attribute modification section 460a fetches the default attribute value from the default profile 460b and modifies the attribute information. By virtue of such a configuration, it becomes possible to modify the attribute information without the specification of the attribute value by the user.

In the seventh embodiment and the modified example thereof, an explanation was only given of the case where the attribute modification section 512e is provided in the job scheduling section 12b. However, the attribution modification section 512e can be provided independently outside the job scheduling section.

Claims

1. A job processing system comprising a terminal equipment for issuing a job request by handling a plurality of documents as one job, and a job scheduling device which sequentially processes jobs by storing the jobs, received from the terminal equipment through a network, in a queue and sending a job execution section a processing request relating to a document specified by the job stored in the queue,

said terminal equipment comprising:

attribute information adding means for adding information which specifies a job output method to a job request as attribute information of the job, and

said job scheduling device comprising:

attribute information setting means for acquiring attribute information included in the received job and sets the attribute information to information which specifies a job and a document;

a queue for storing, as a job, a group of items of the information which specify a job and a document; and

output result control means which, upon reference to the information items which specify a job and a document with respect to the job

wait to information for specifying this leading document; and

control state setting means which, if the processing completion wait is set to information which specifies the leading document of the job stored in said queuing means, renders that job in a processing completion wait state, wherein said job scheduling device sequentially retrieves jobs stored in said queuing means when the job execution section becomes enabled to accept processing, issues a processing request for a corresponding document when there is information specifying a document to which a processing request can be issued, and when a job is placed in the processing completion wait state, prevents the issue of processing requests with respect to a document for that job and documents for subsequent jobs until that job is released from the processing completion wait state by a user's instruction or a timeout.

5. A job processing system comprising a terminal equipment for issuing a processing request by handling a plurality of documents as one job, a job execution section for printing the documents, and a job scheduling device which accepts a document input from the terminal equipment through a network and issues a processing request relating to that document to the job execution section,

said terminal equipment comprising:
control information setting means for specifying a password input wait for a leading document among the plurality of documents, and
said job scheduling device comprising:
preparation means for preparing information which specifies a received document;
queuing means for storing the information which specifies the document by associating the information on a job-by-job basis;
control information setting means which, if a password input wait is set for the leading document among a plurality of received documents, sets the password input wait to information which specifies that leading document; and
control state setting means which, if the password input wait state is set to information which specifies the leading document of the job stored in the queuing means, renders that job in a password input wait state, wherein said job scheduling device sequentially retrieves jobs stored in said queuing means when the job execution section becomes enabled to accept processing, issues a processing request for a corresponding document when there is information specifying a document to which a processing request can be issued, and when a job is placed in the

password input wait state, prevents the issue of processing requests with respect to a document of that job and documents of subsequent jobs until that job is released from the password input wait state by a user's instruction or a timeout.

6. A job processing system comprising a terminal equipment for issuing a processing request, and a job scheduling device which sequentially processes jobs by storing the jobs received from the terminal equipment in a queue and issuing a processing request, relating to a document specified by the job stored in the queue, to a job execution section,

said terminal equipment comprising:
attribute information adding means for adding information relating to job wait control and message information relating to the wait control to the job request as attribute information, and
said scheduling device comprising:
job information preparing means for preparing job information which specifies a received job;
attribute information setting means for setting attribute information included in the received job in the job information;
a queue for storing the prepared job information in order;
control state setting means which, if wait control is set to the job information stored in said queue, renders a job associated with that job information in a wait control state when processing of that job is started or completed; and
message information informing means which, when the job enters the wait control state, informs said terminal equipment of message information set with respect to that job.

7. A print processor which prints a document in response to a job request received through a network, said print processor comprising:

job accepting means for accepting the job request;
queuing means for storing the accepted jobs in sequential order;
output means for printing a document specified by the job stored in said queuing means;
converting means for converting the document into a format which said output means can interpret; and
conversion control means for causing said converting means to convert a document which needs to be converted when being printed by said output means, wherein said converting means informs said conversion control means that it can carry out

to a received print document every time a print document is received;

executing a job by handling a print document received immediately before the judgement of the reception of the next print document as a final print document of this job when the next print document is not received within a predetermined period of time, if the scheduling attributes are the after-complete attributes; and completing the job by dealing a print document received immediately before the judgement of the reception of the next print document as the final document of this job when the next print document is not received within a predetermined period of time, if the scheduling attributes are the before-complete attributes.

13. A job processing device which sequentially executes jobs for which processing requests were accepted, said job processing device comprising:

first queuing means for sequentially storing jobs for which processing requests were accepted;

second queuing means for sequentially storing jobs whose processing is to be interrupted from among the jobs stored in said first queuing means;

queue control means which moves the job stored in the first queuing means from said first queuing means to said second queuing means in response to a job processing interruption request and moves the jobs stored in said second queuing means from said second queuing means to said first queuing means in response to a job resumption request; and

job execution means for sequentially executing the jobs stored in said first queuing means.

14. The job processing device as defined in claim 13, further comprising job state control means which manages interruption information, representing an interrupted state of a job, as attribute information of the job when the job being executed by said job execution means is interrupted in response to the job processing interruption request, wherein the execution of the job is resumed with reference to the interruption information in response to the job resumption request.

15. A job scheduling device which sequentially stores jobs, for which processing requests were received from terminals, in a queue and sequentially processes the jobs held in the queue using a job execution section, said job scheduling device comprising:

a plurality of queues provided corresponding to states of the jobs;

scheduling means for scheduling the jobs using the plurality of queues; and

recovery means for recovering previous state of each of the jobs being held in the plurality of queues, at the time of recovery from a failure, if any failure occurred while the jobs are being scheduled by said scheduling means.

16. A job scheduling device for storing, in a queue, print jobs which include print data and attribute information and for which processing requests were received from terminals, and for sequentially printing the print jobs held in the queue based on the attribute information using a job execution section, said job scheduling device comprising:

a plurality of queues provided corresponding to print job states;

scheduling means for scheduling the print jobs using the plurality of queues; and

attribute modifying means for modifying the attribute information only when a print job can be changed at the time that an instruction for modifying the attribute information of the print job is received, and when the instruction is free from errors.

FIG. 1

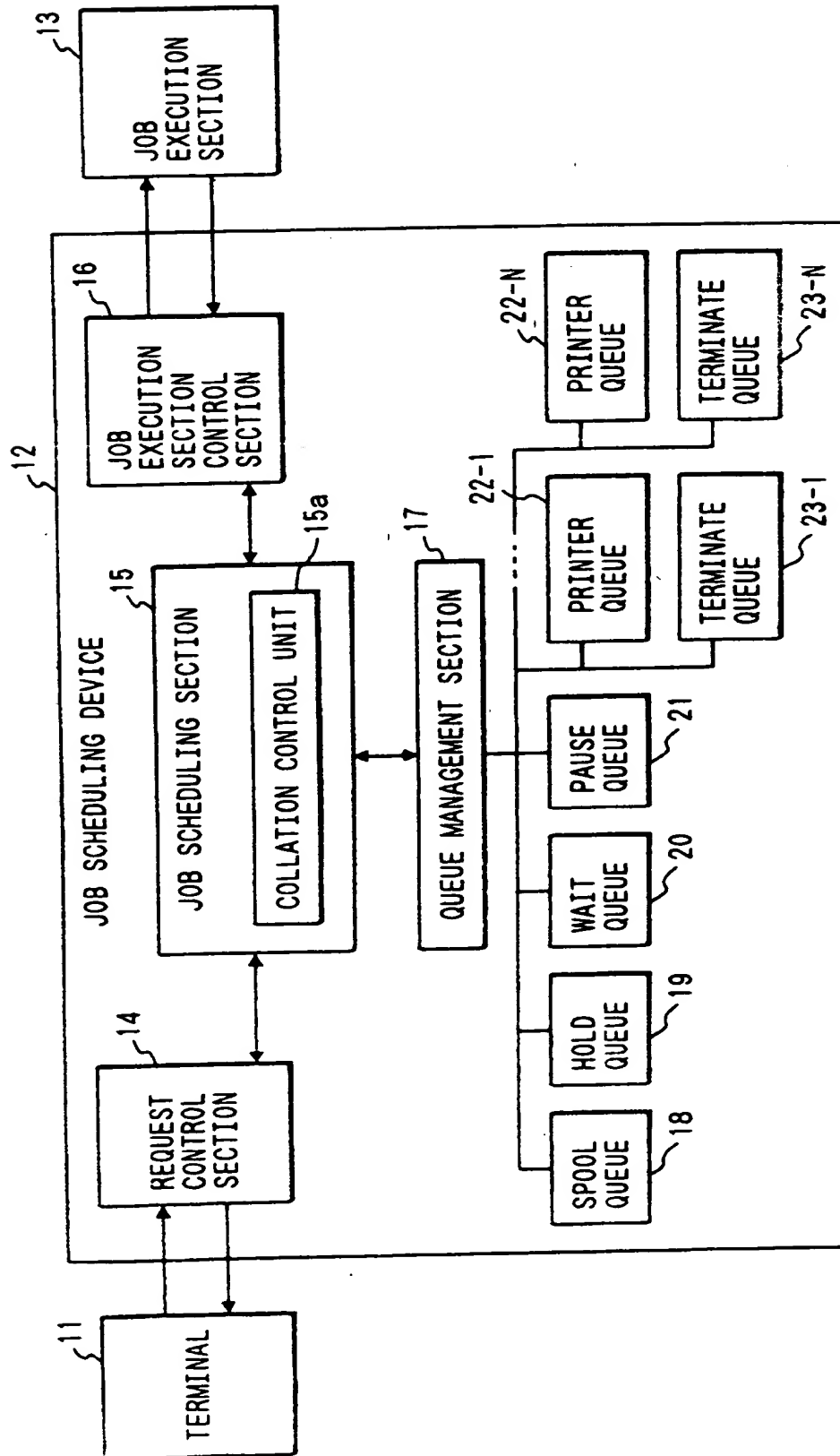


FIG. 2

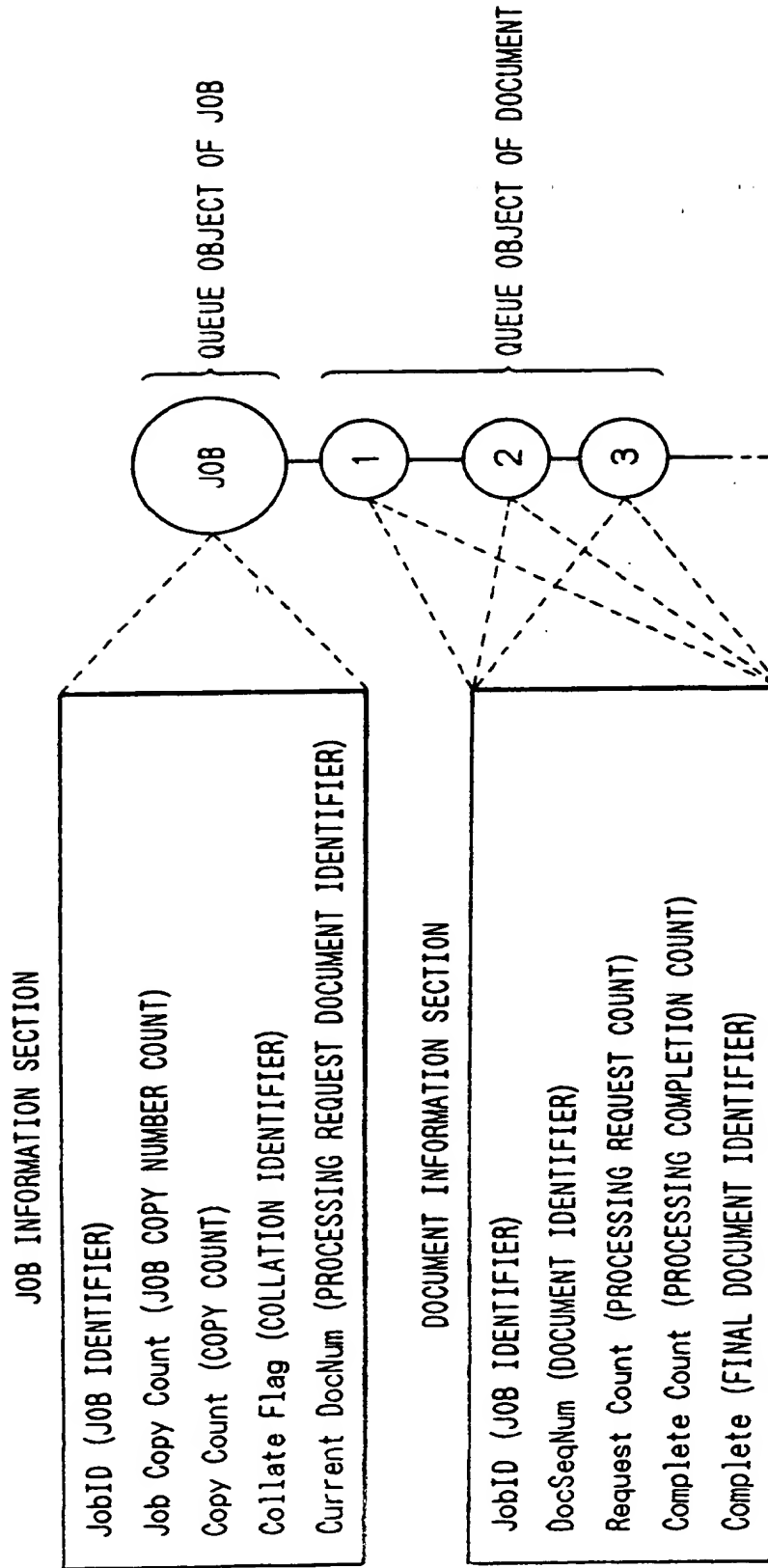


FIG. 3

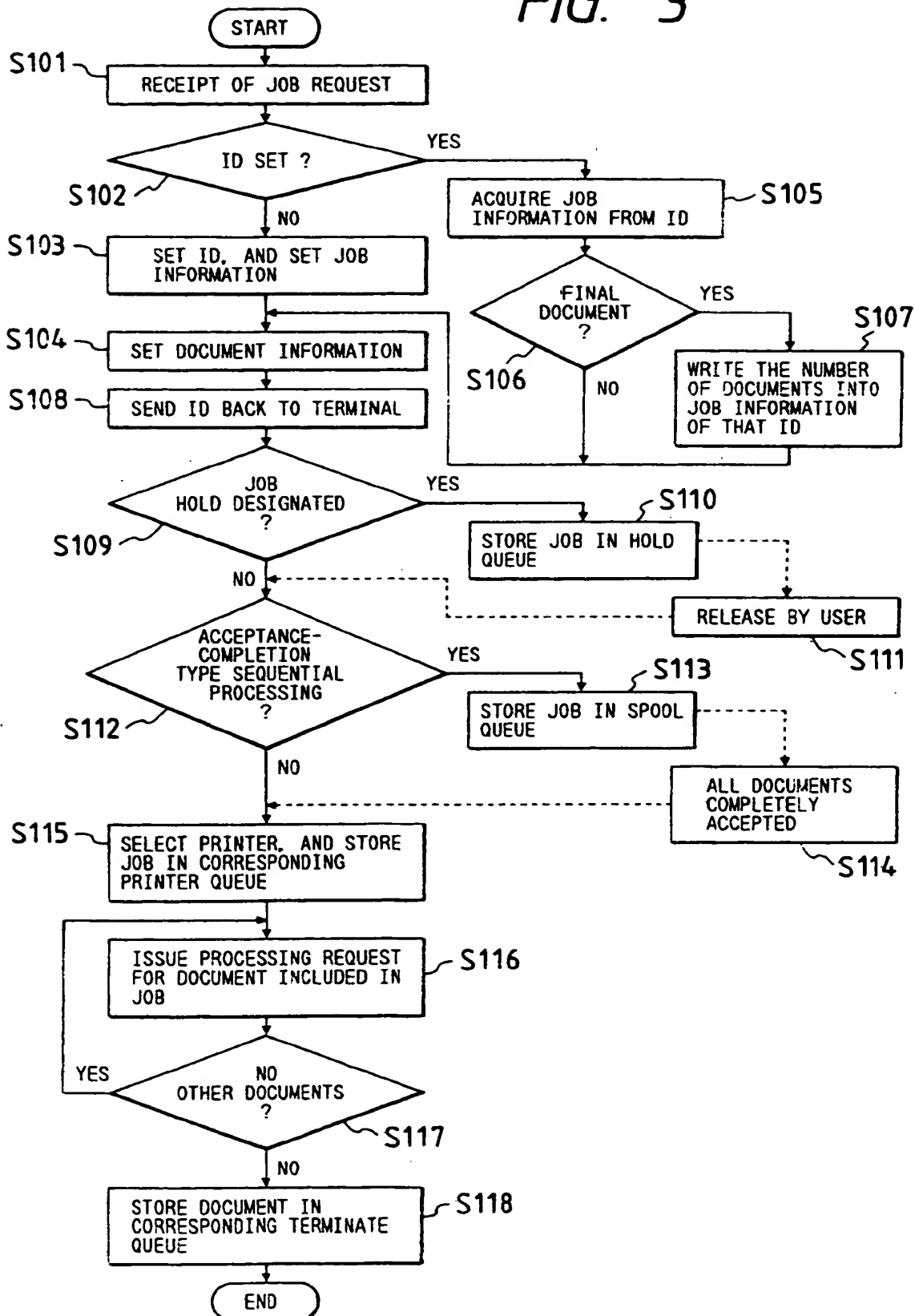


FIG. 4

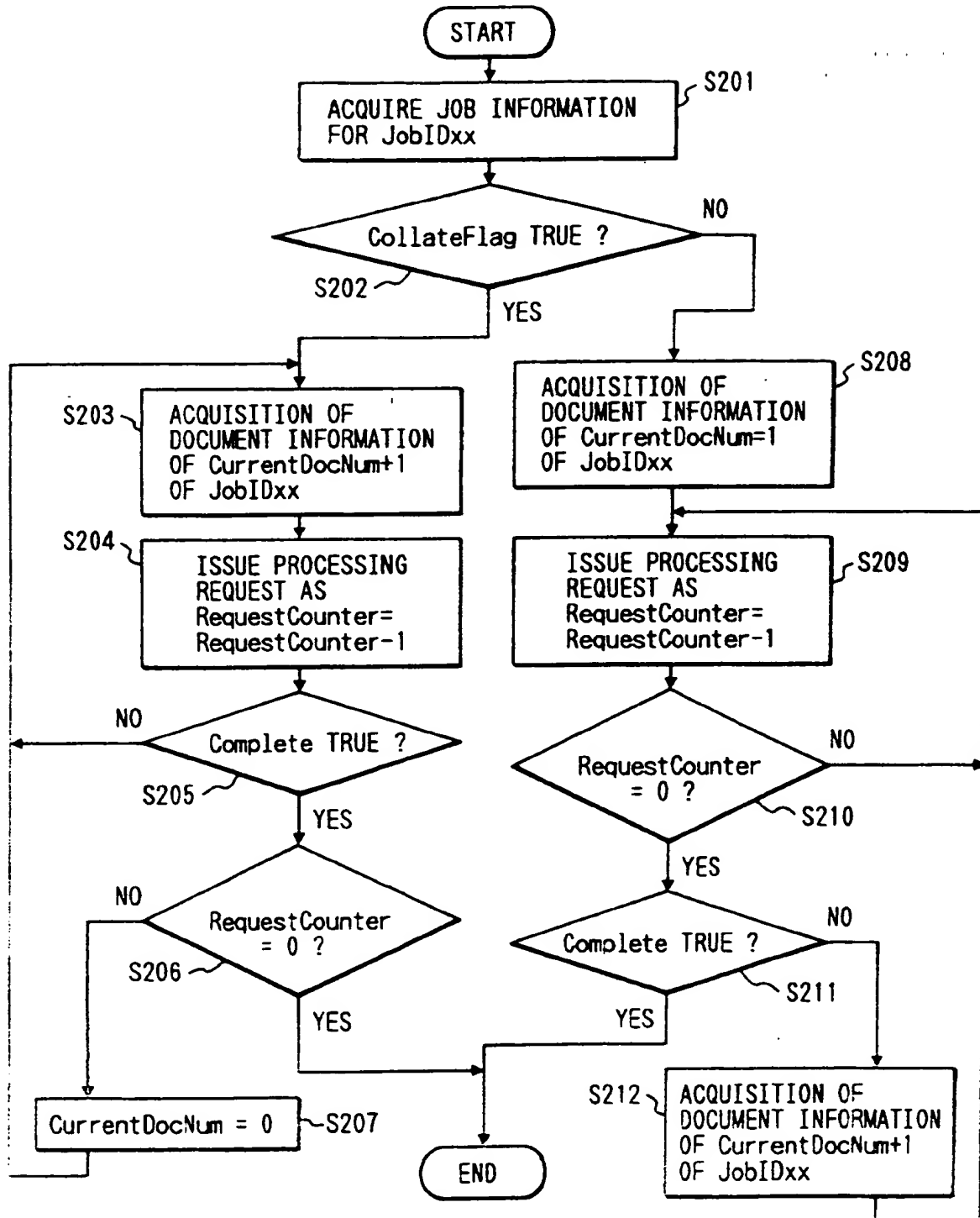
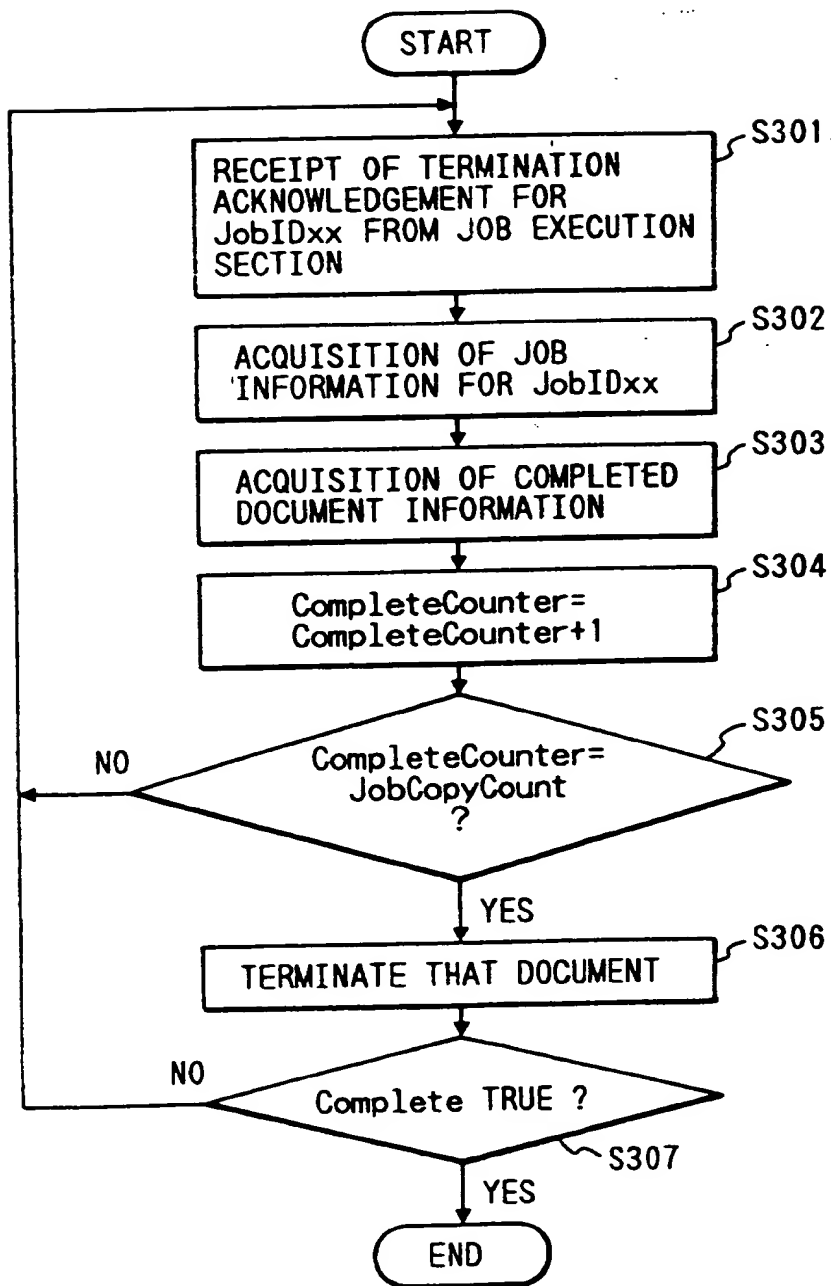


FIG. 5



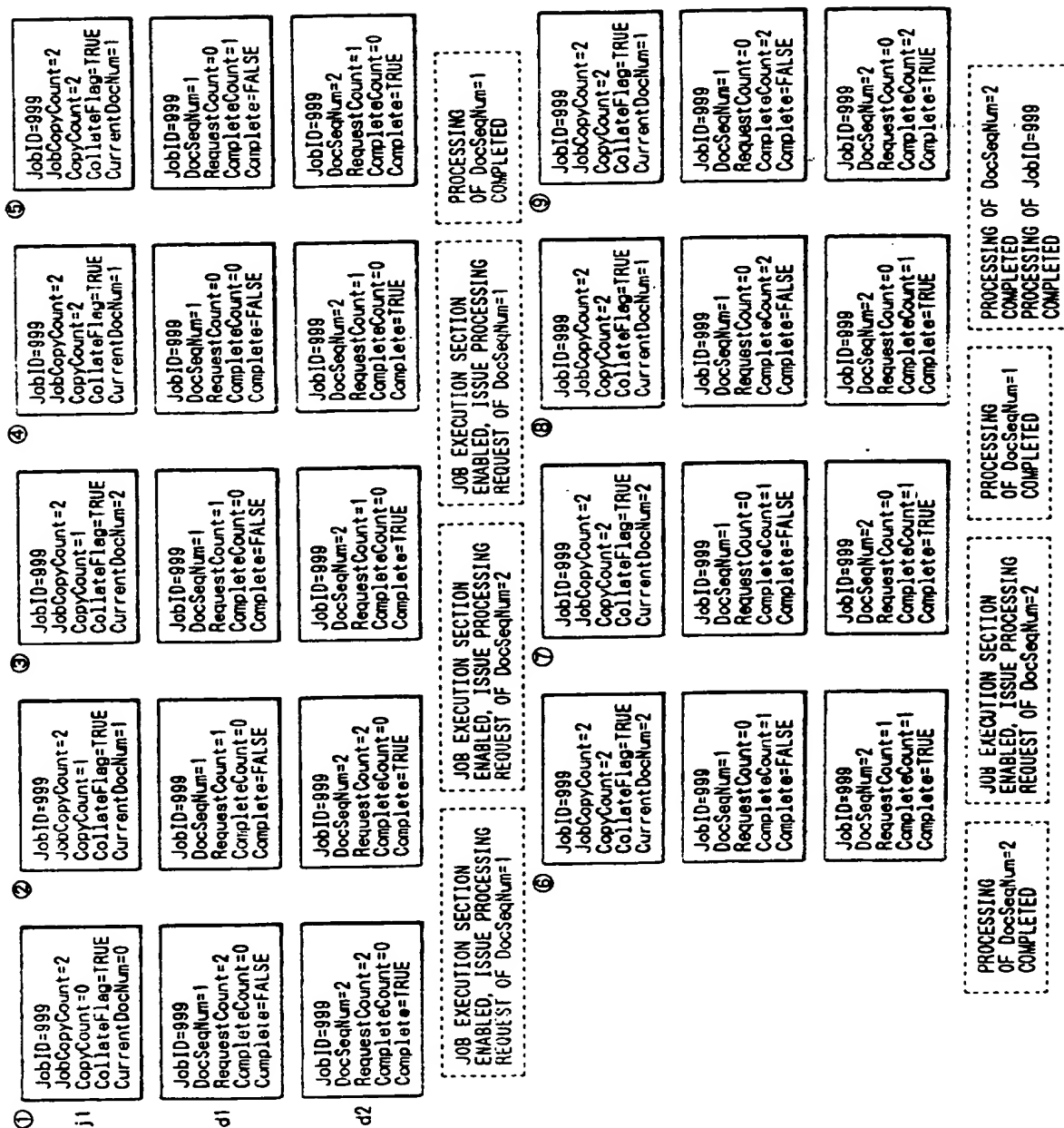


FIG. 6

FIG. 7

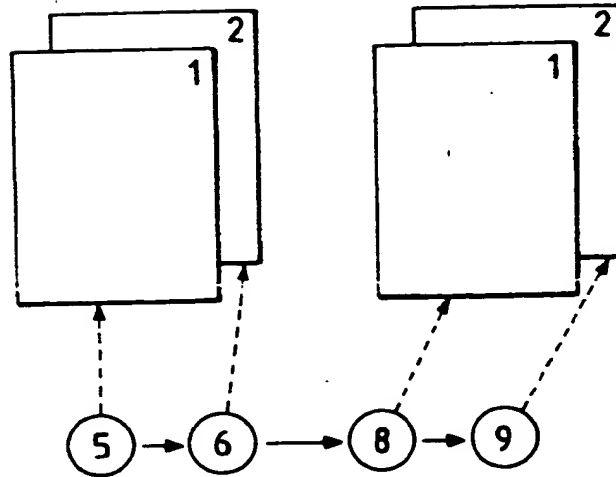
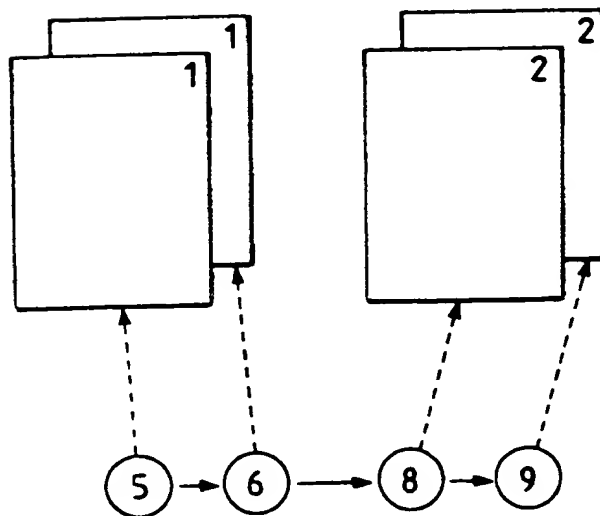


FIG. 9



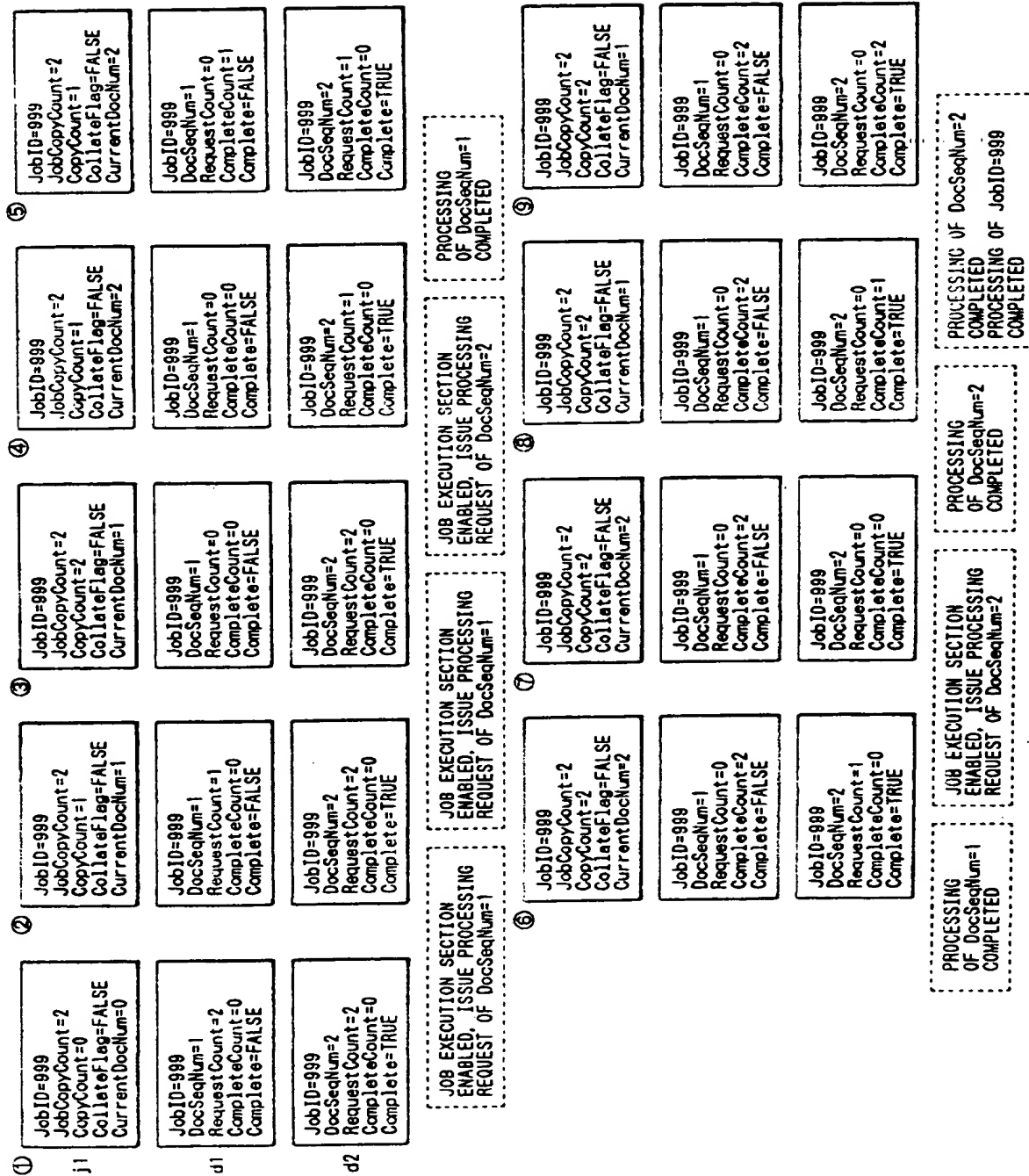


FIG. 8

FIG. 10

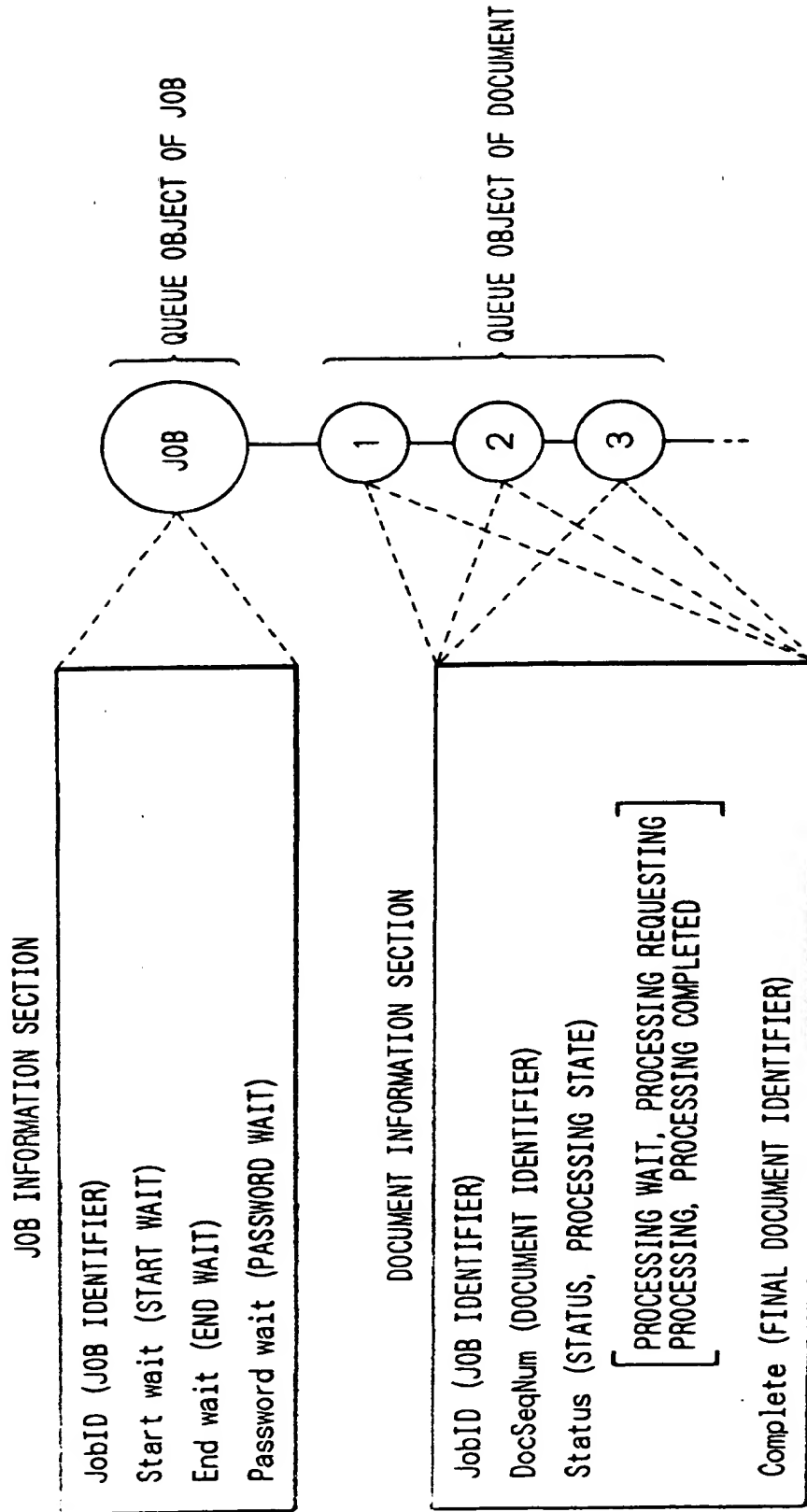


FIG. 11

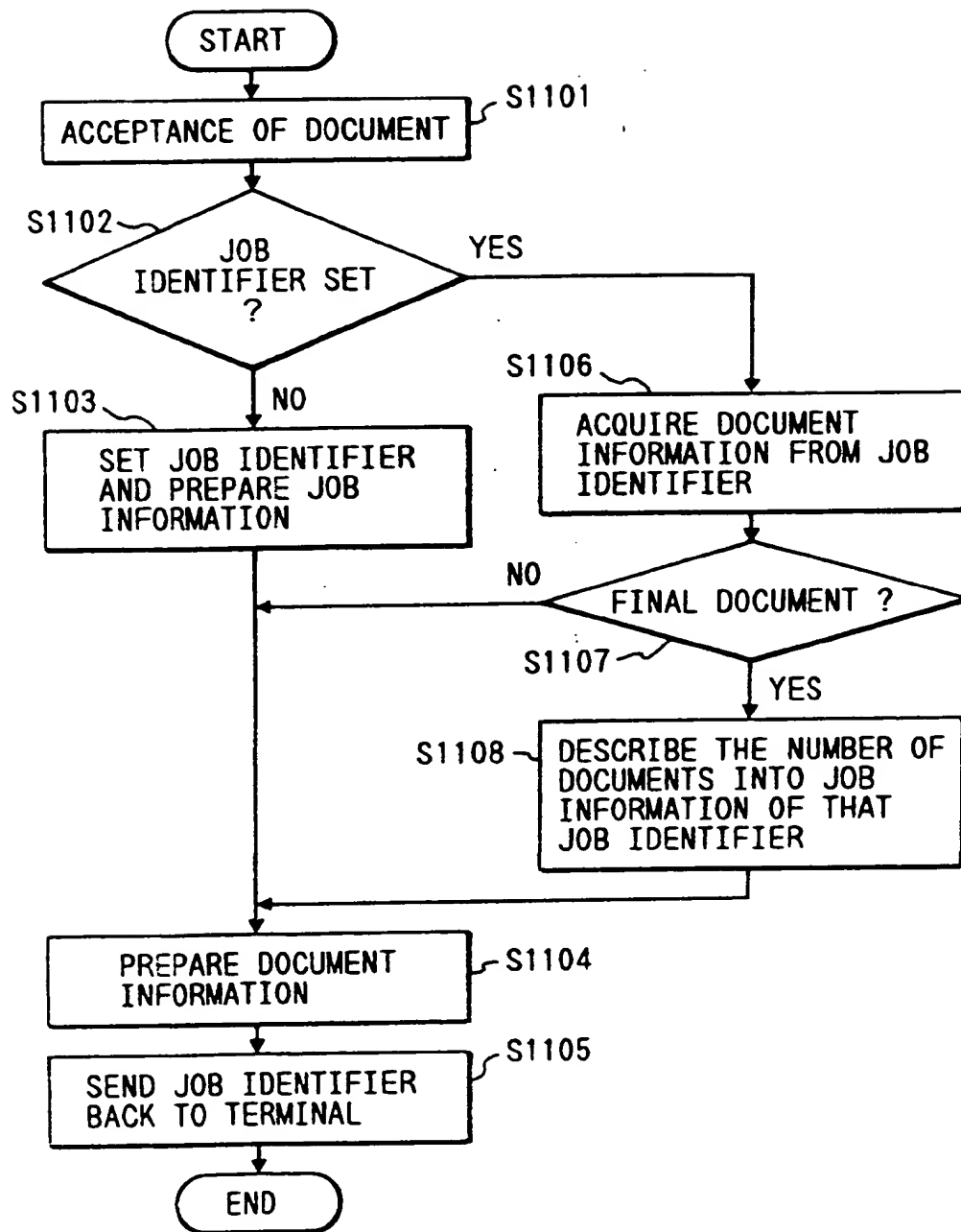


FIG. 12

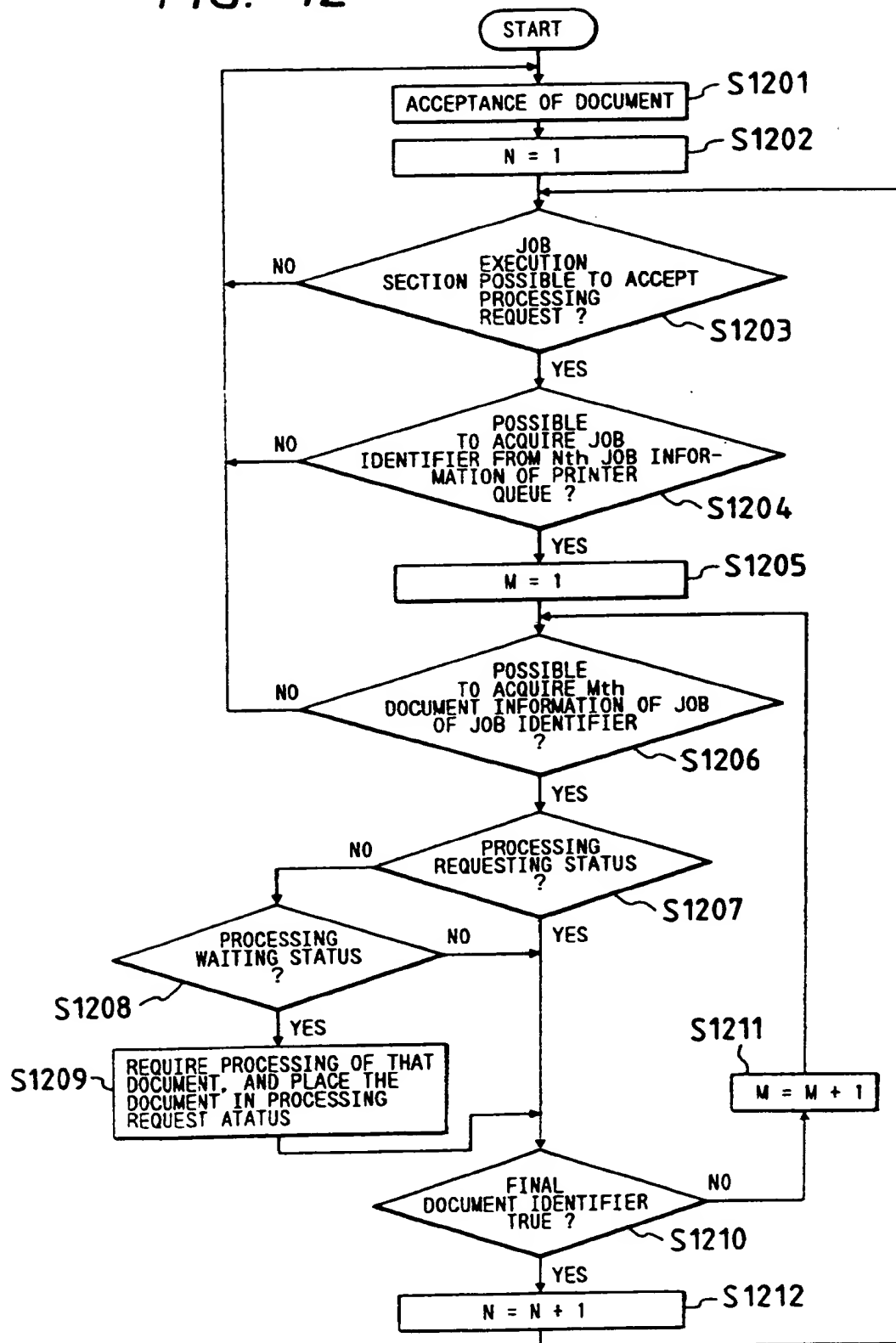


FIG. 13

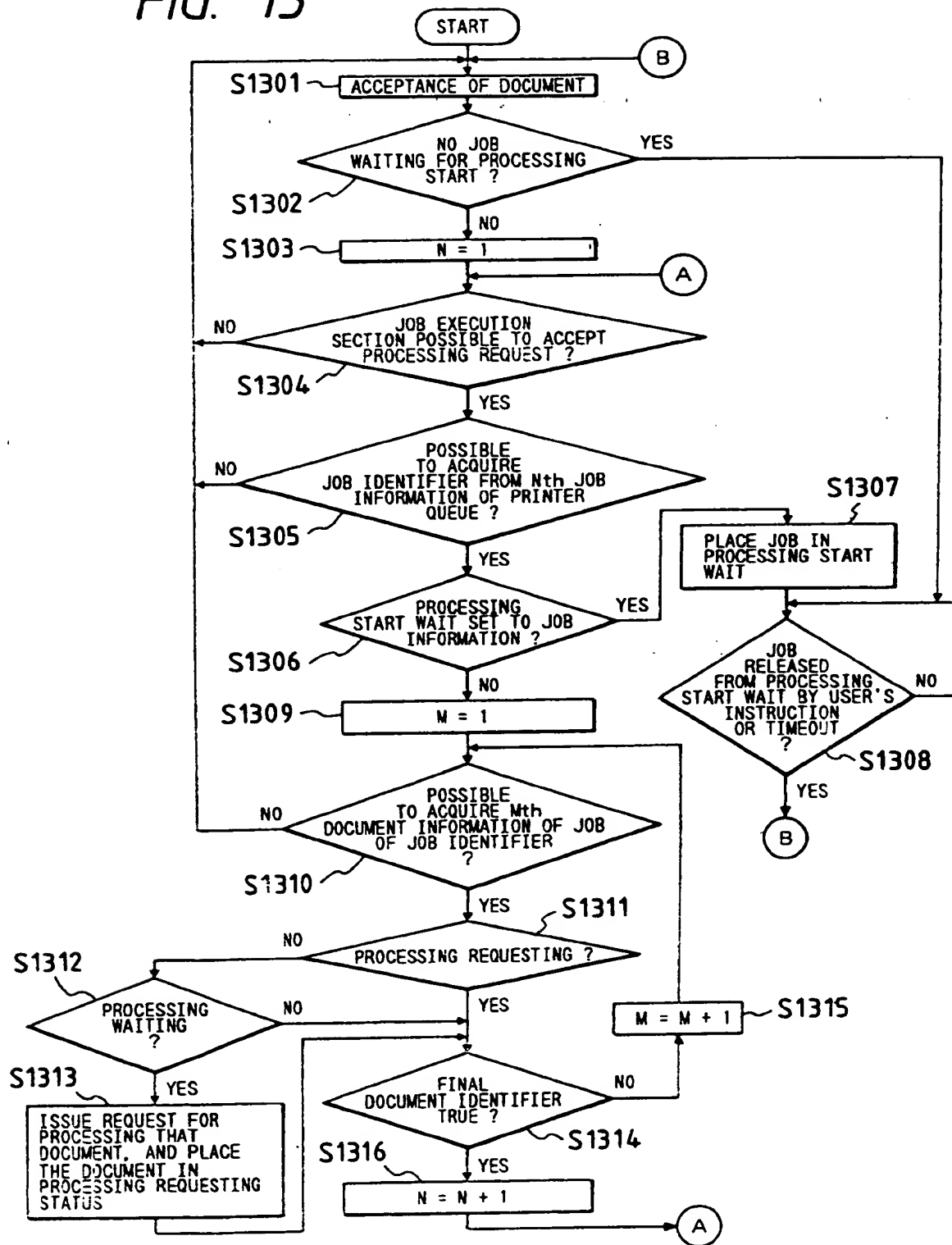


FIG. 14

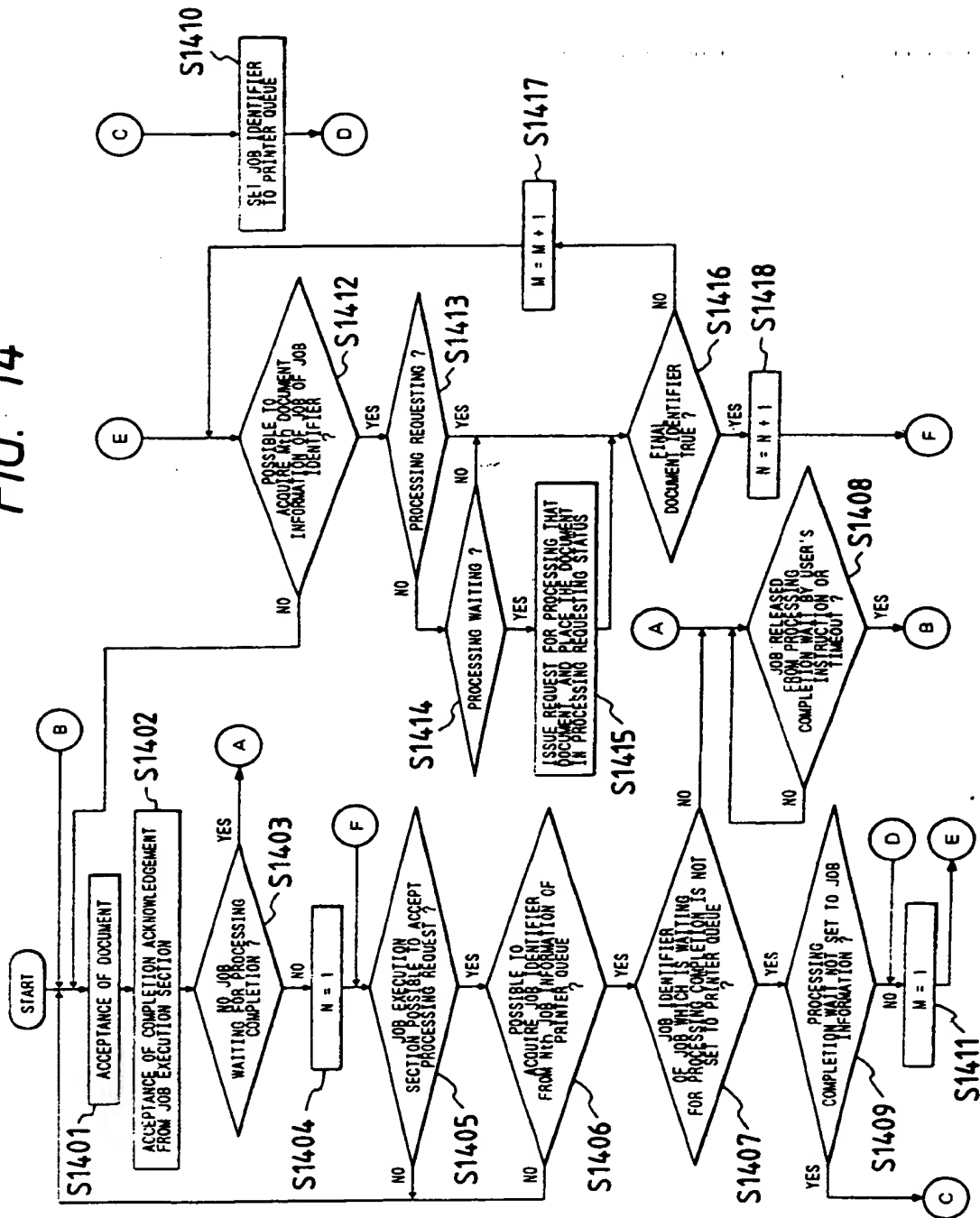


FIG. 15

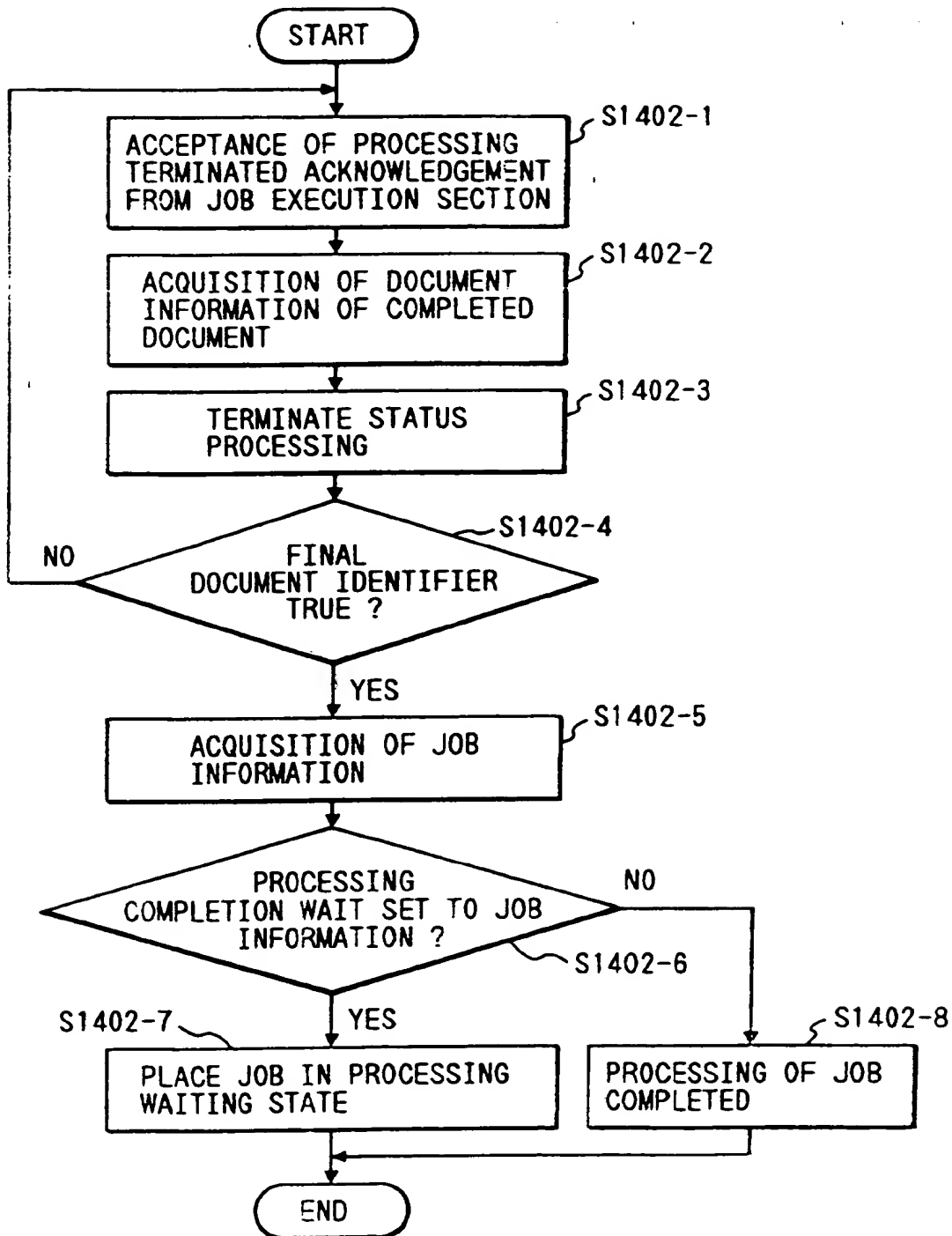


FIG. 16

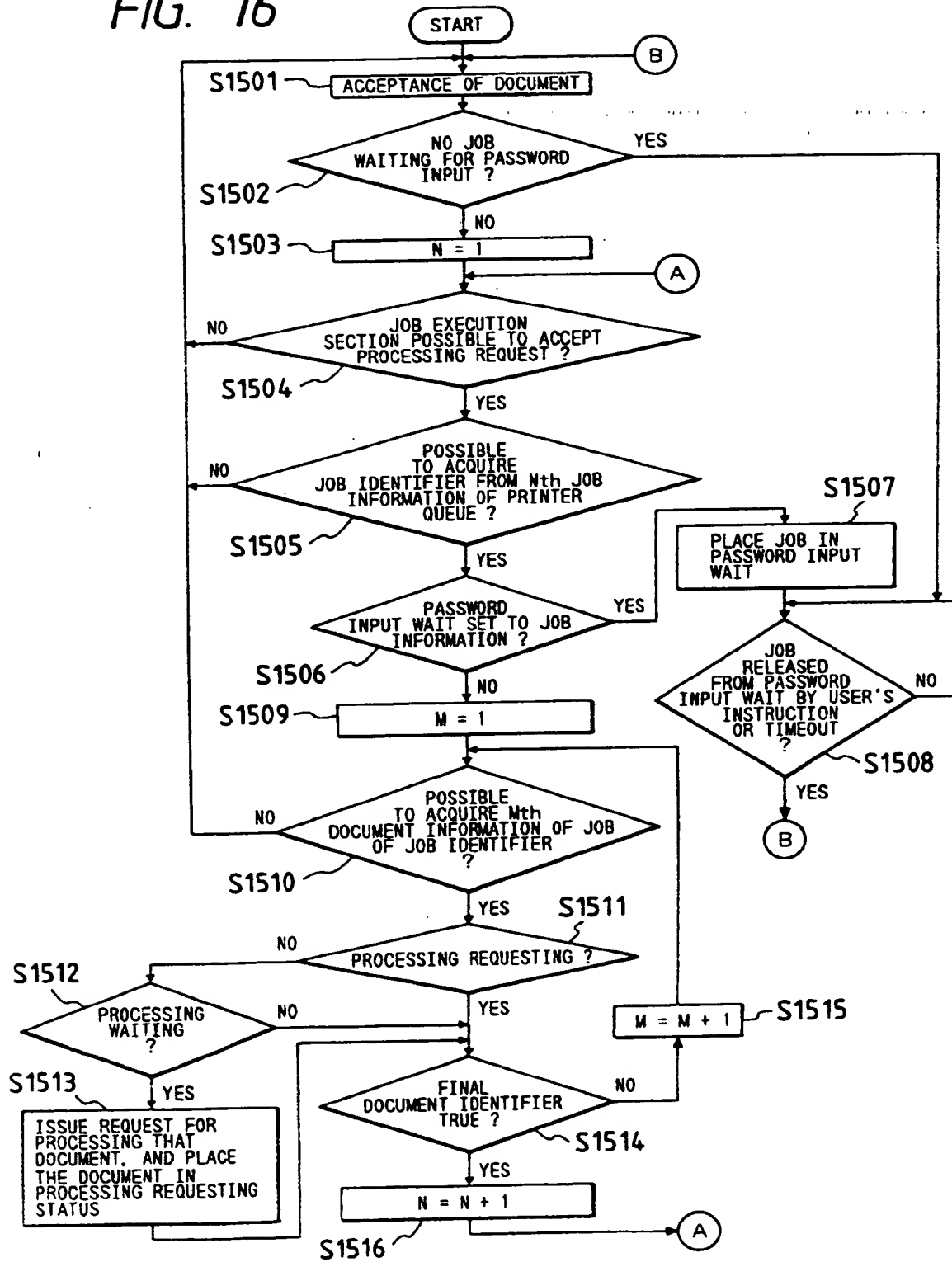


FIG. 17

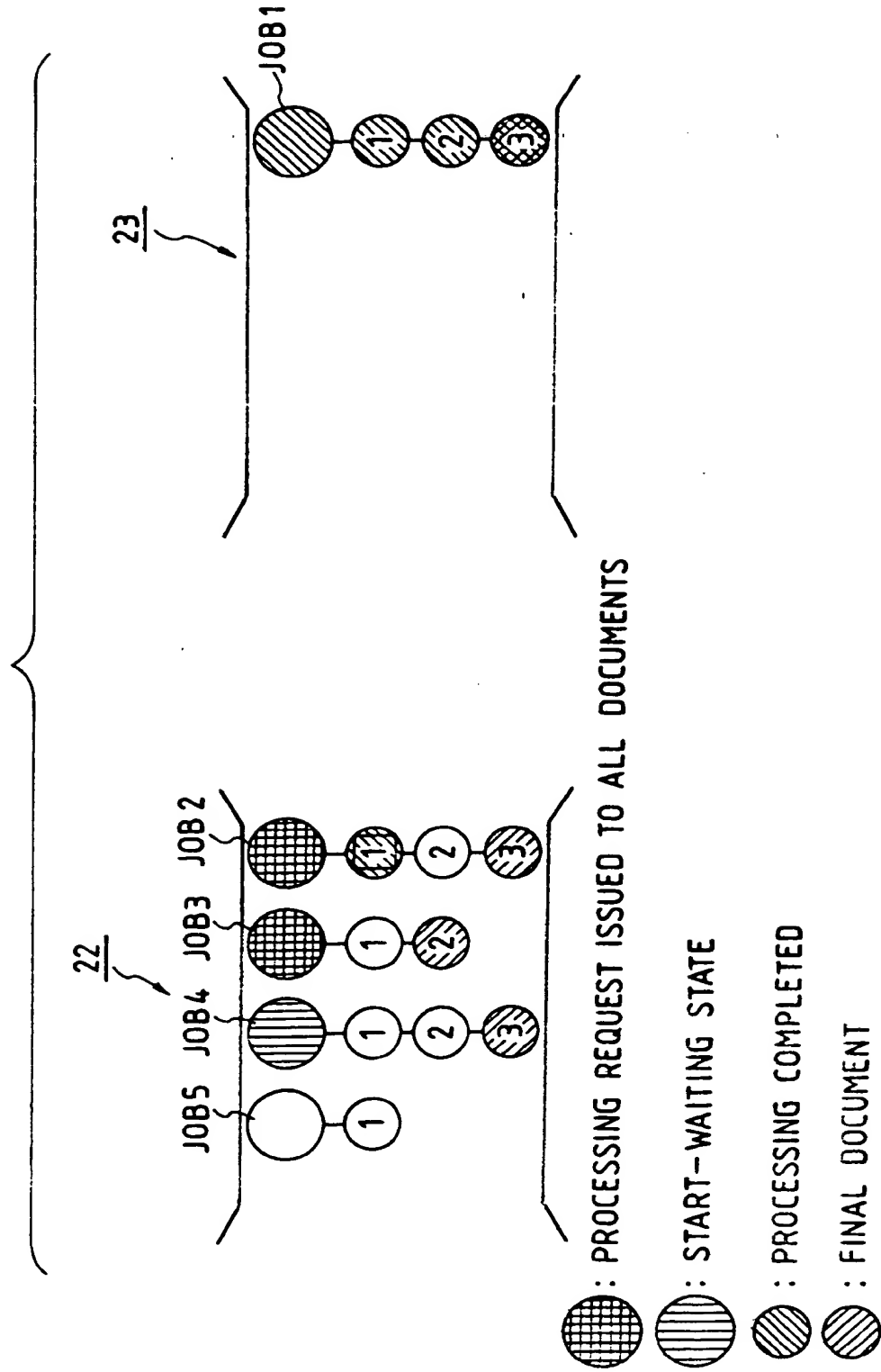


FIG. 18

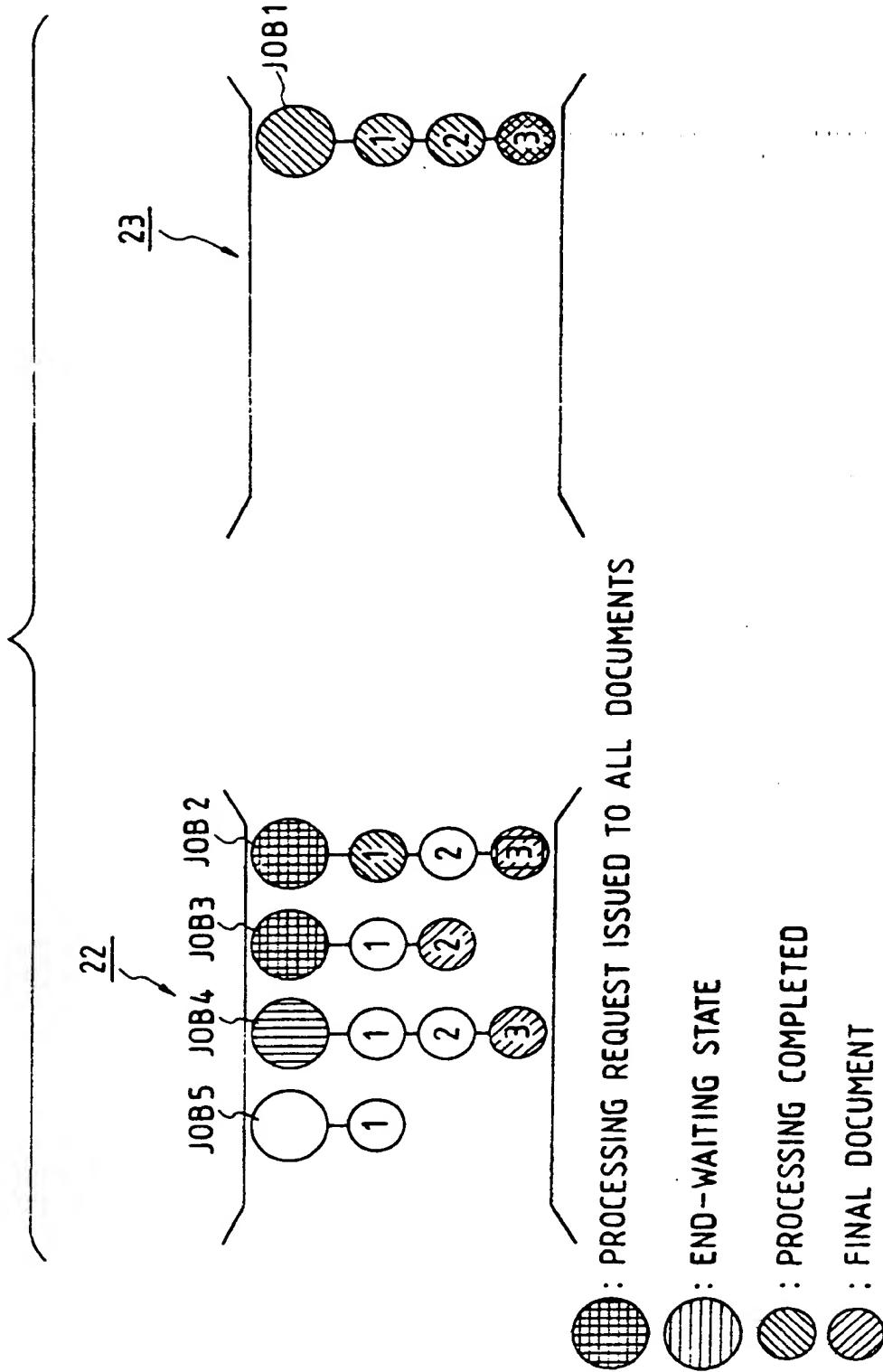


FIG. 19

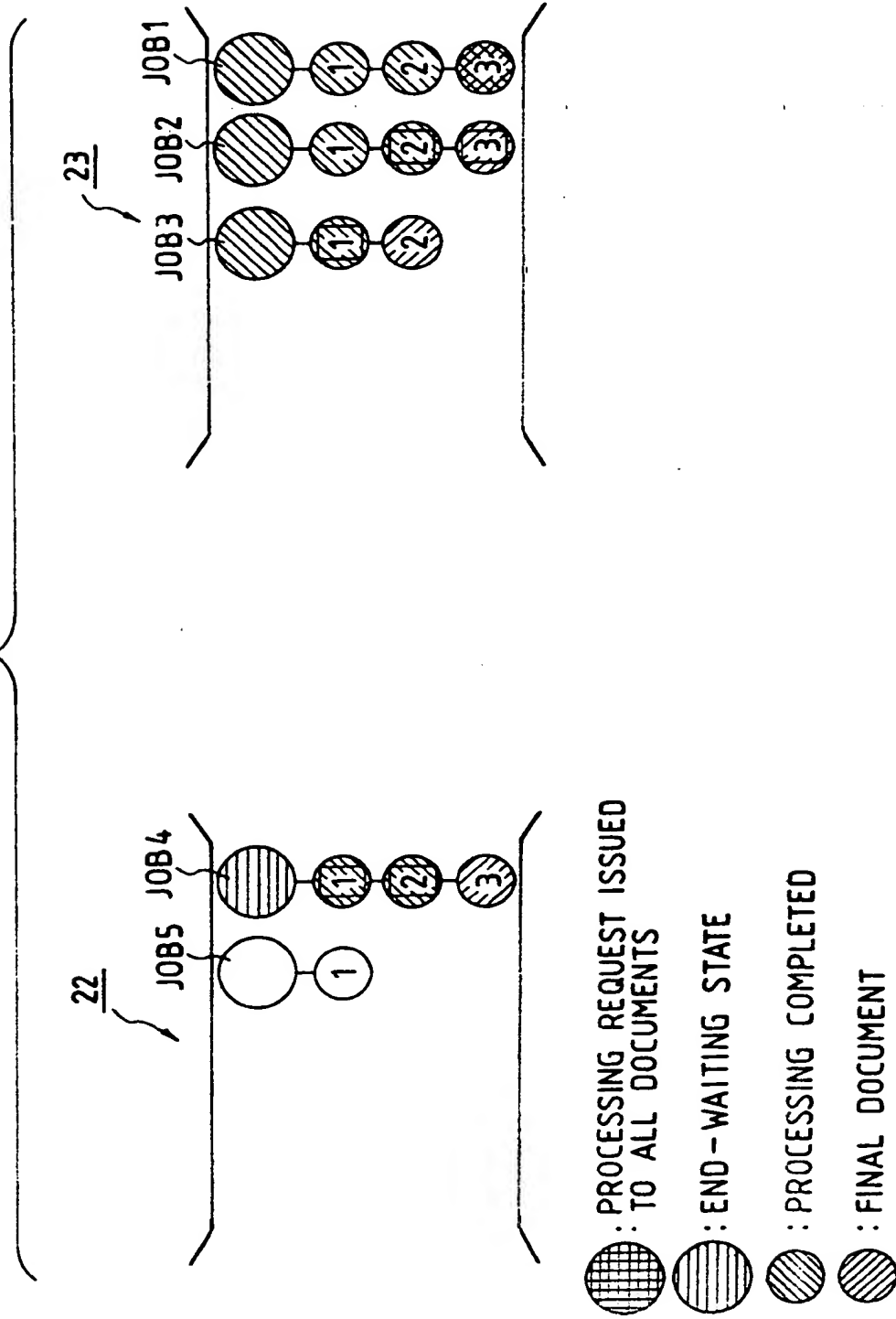


FIG. 20

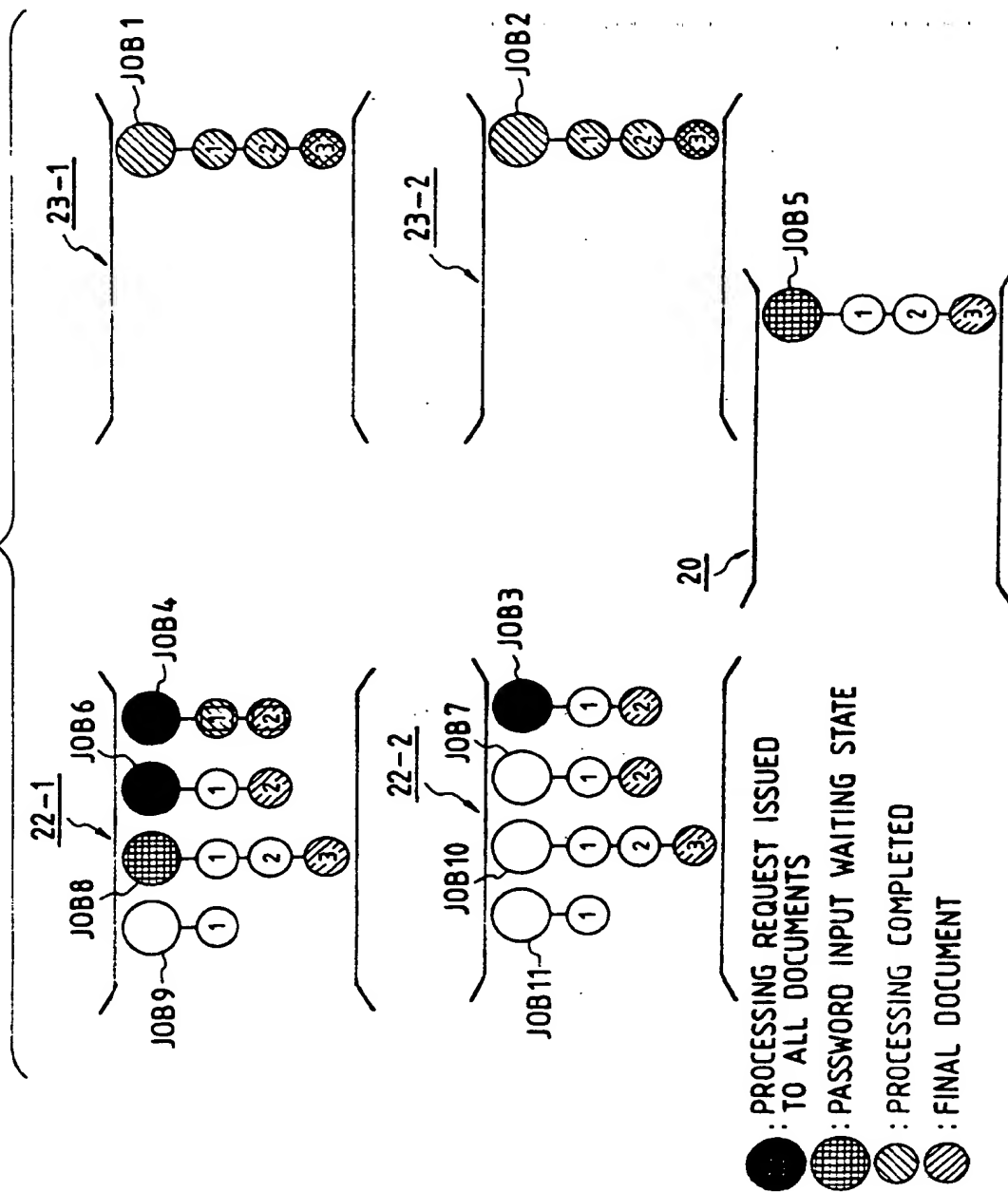
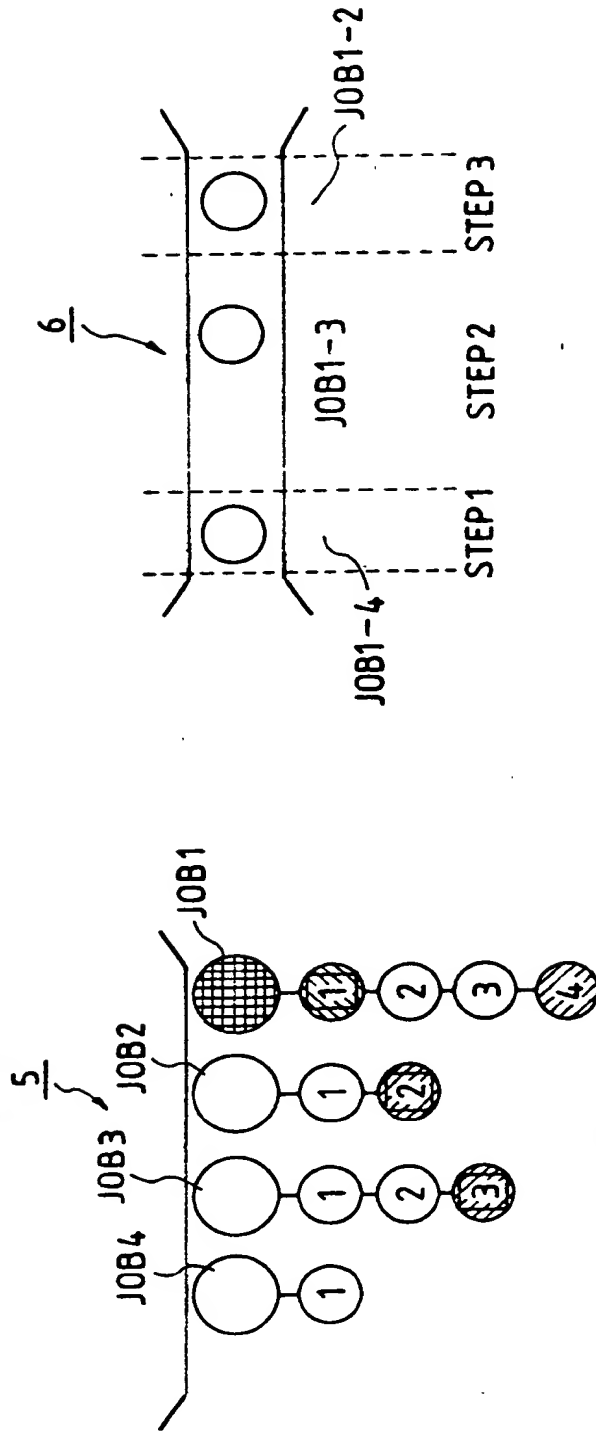


FIG. 21 PRIOR ART






-  : PROCESSING REQUEST ISSUED TO ALL DOCUMENTS
-  : PROCESSING COMPLETED
-  : FINAL DOCUMENT

FIG. 22

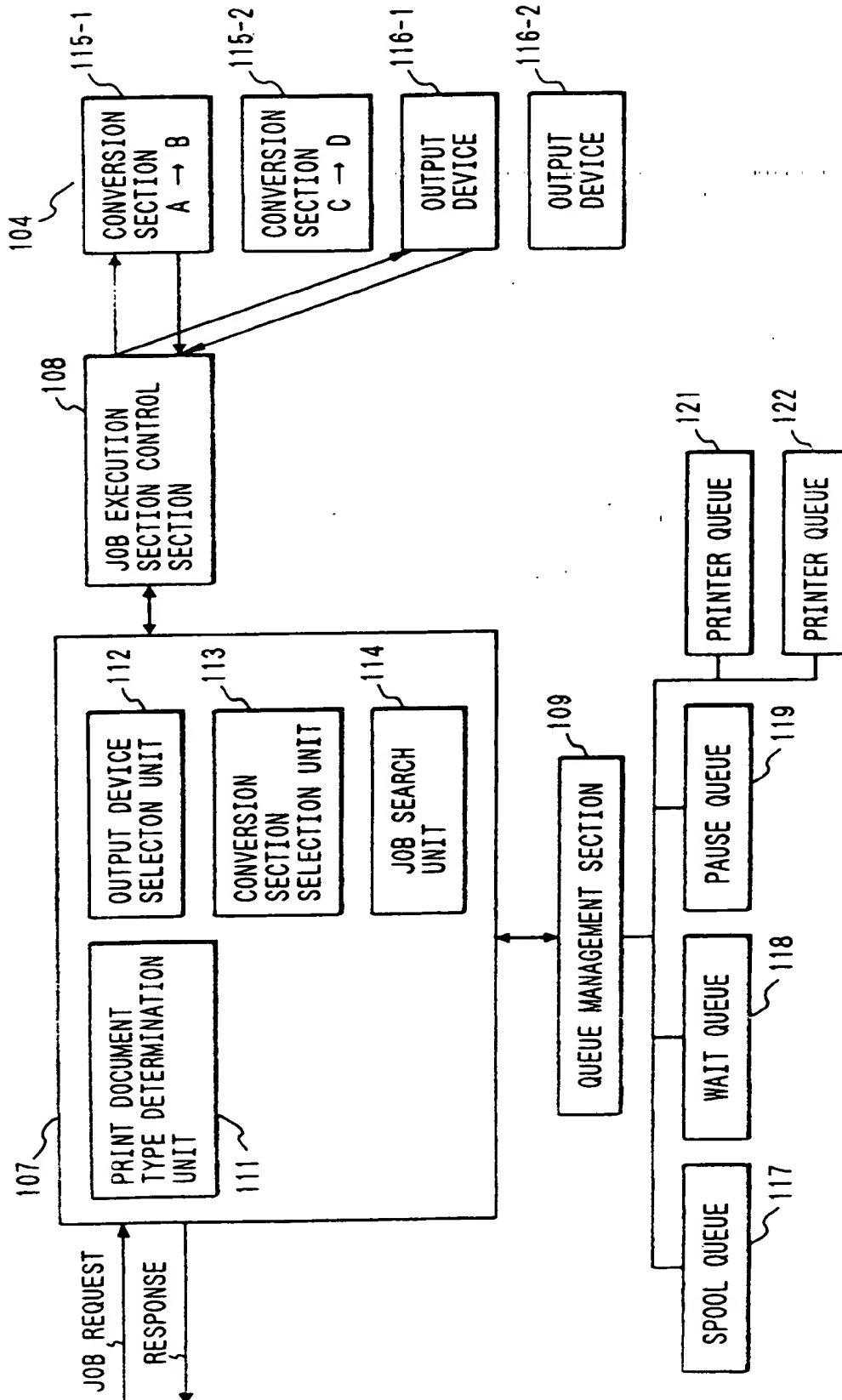


FIG. 23

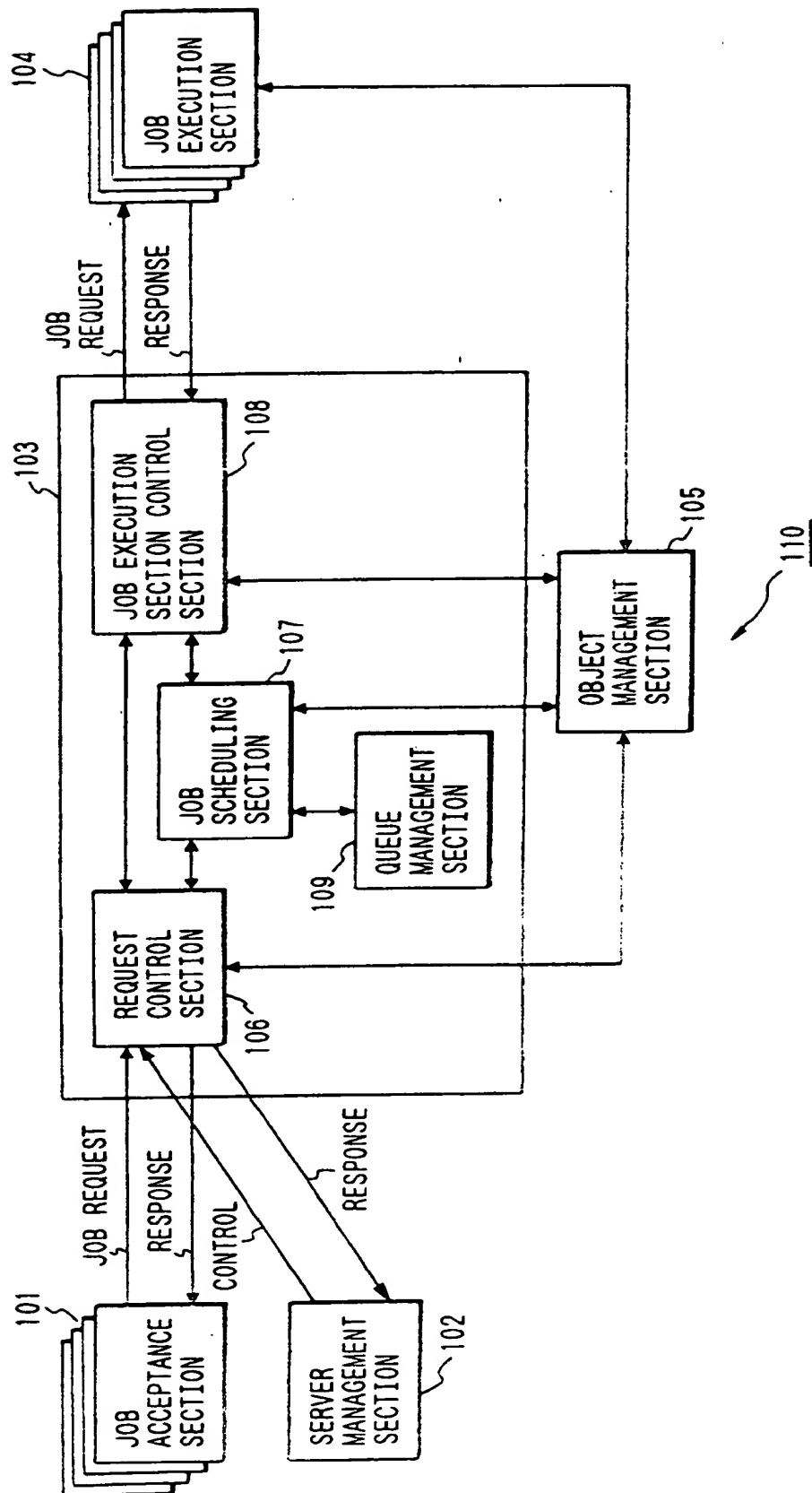


FIG. 24

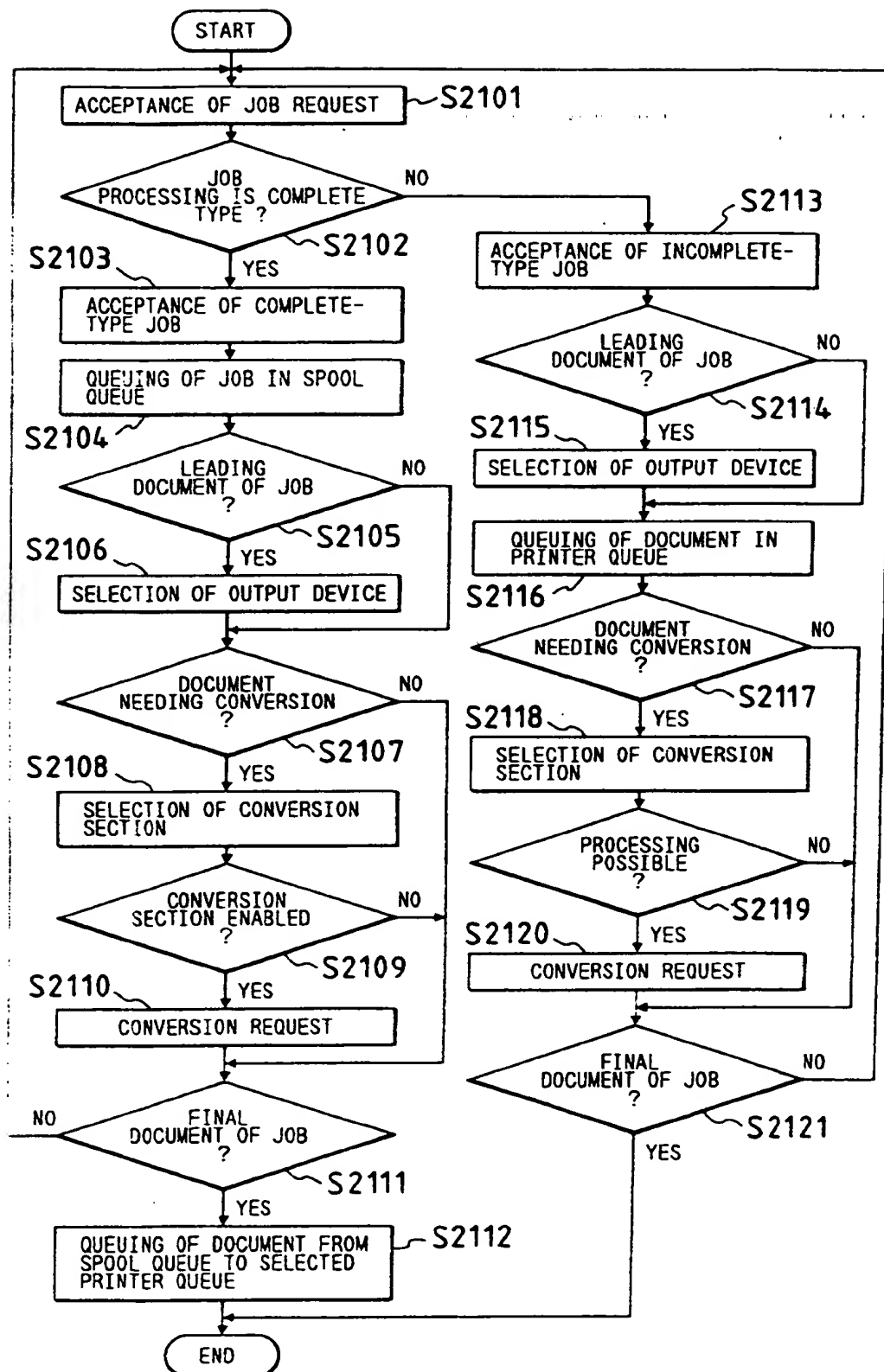


FIG. 25

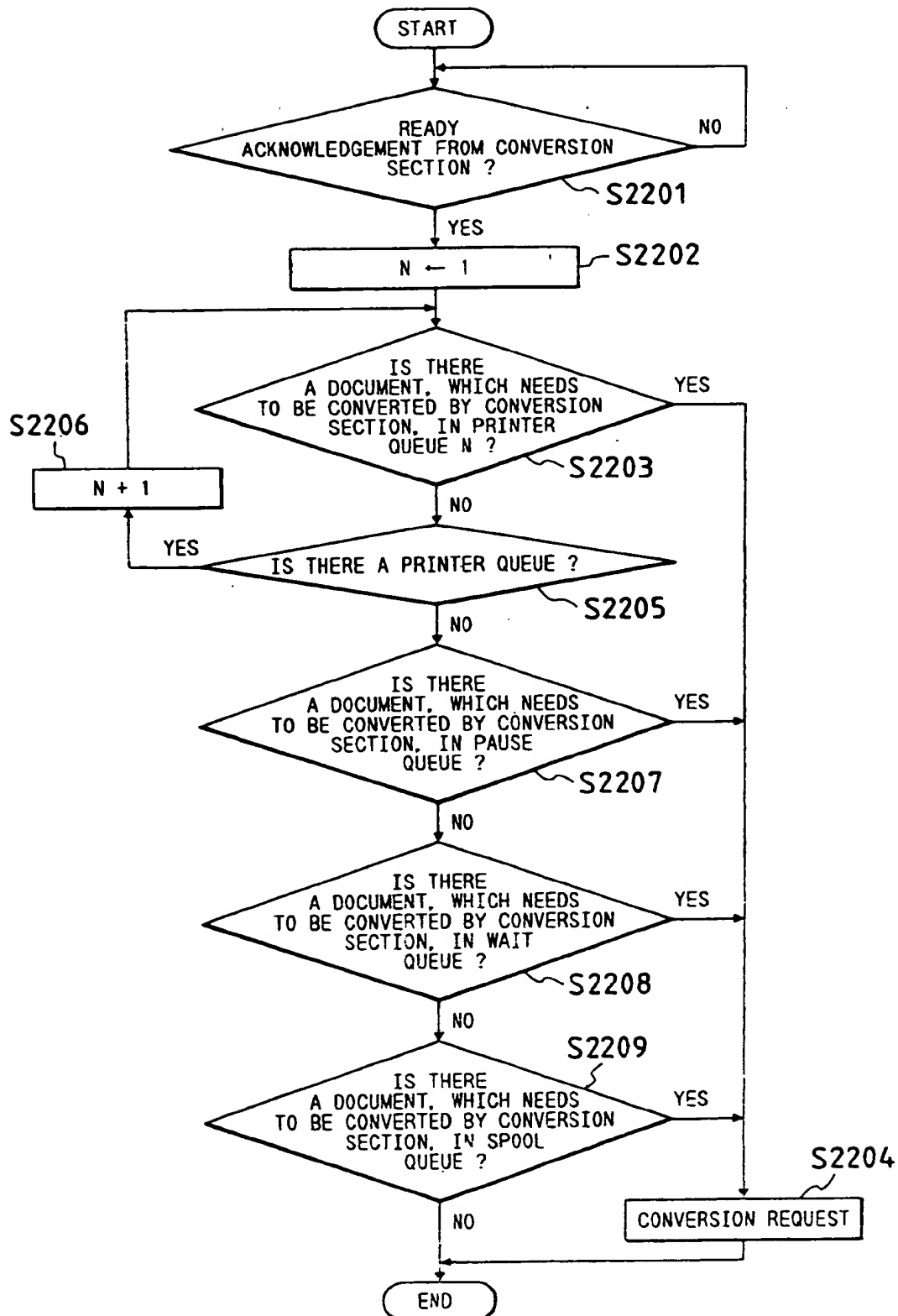


FIG. 26

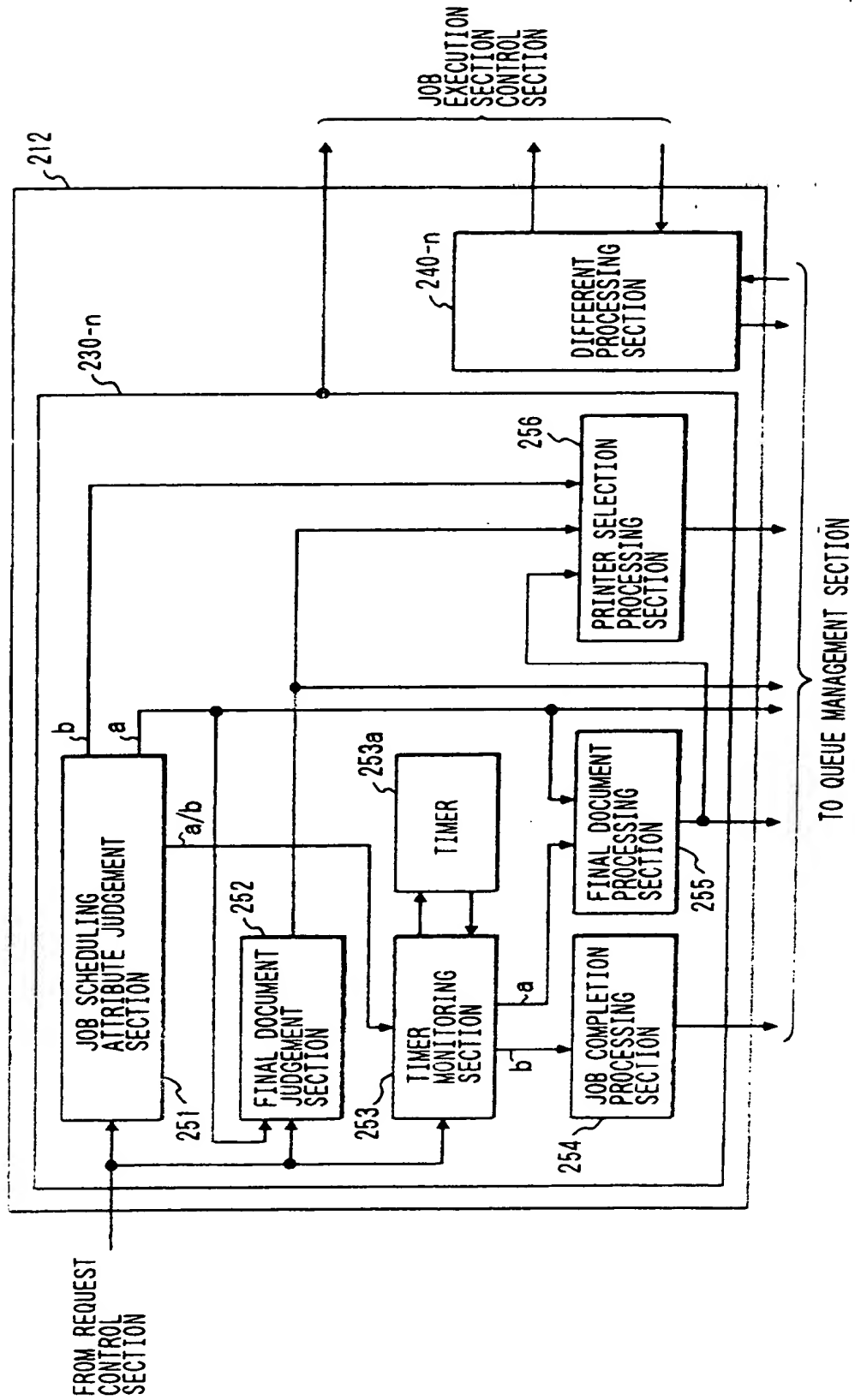


FIG. 27

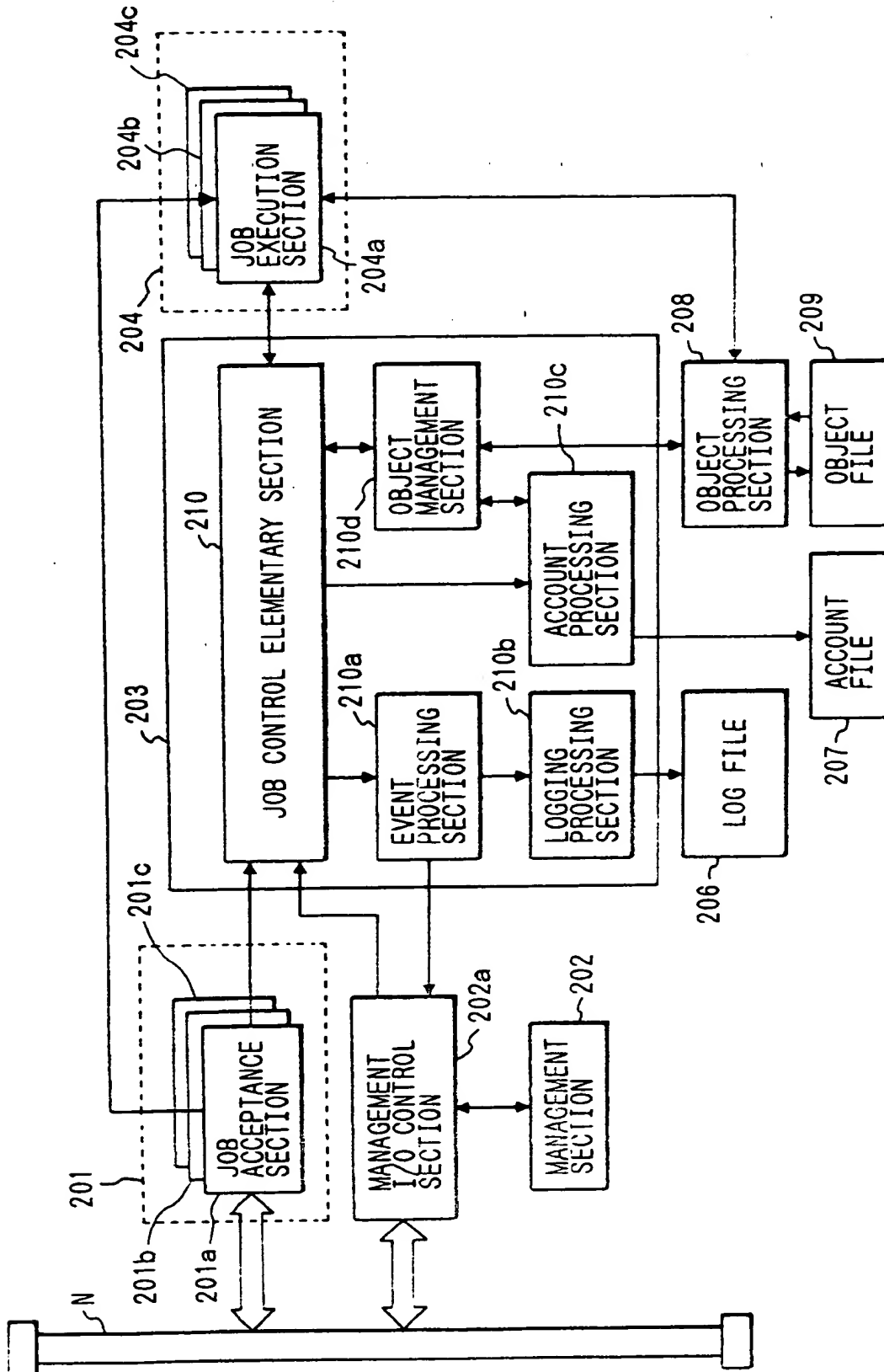


FIG. 28

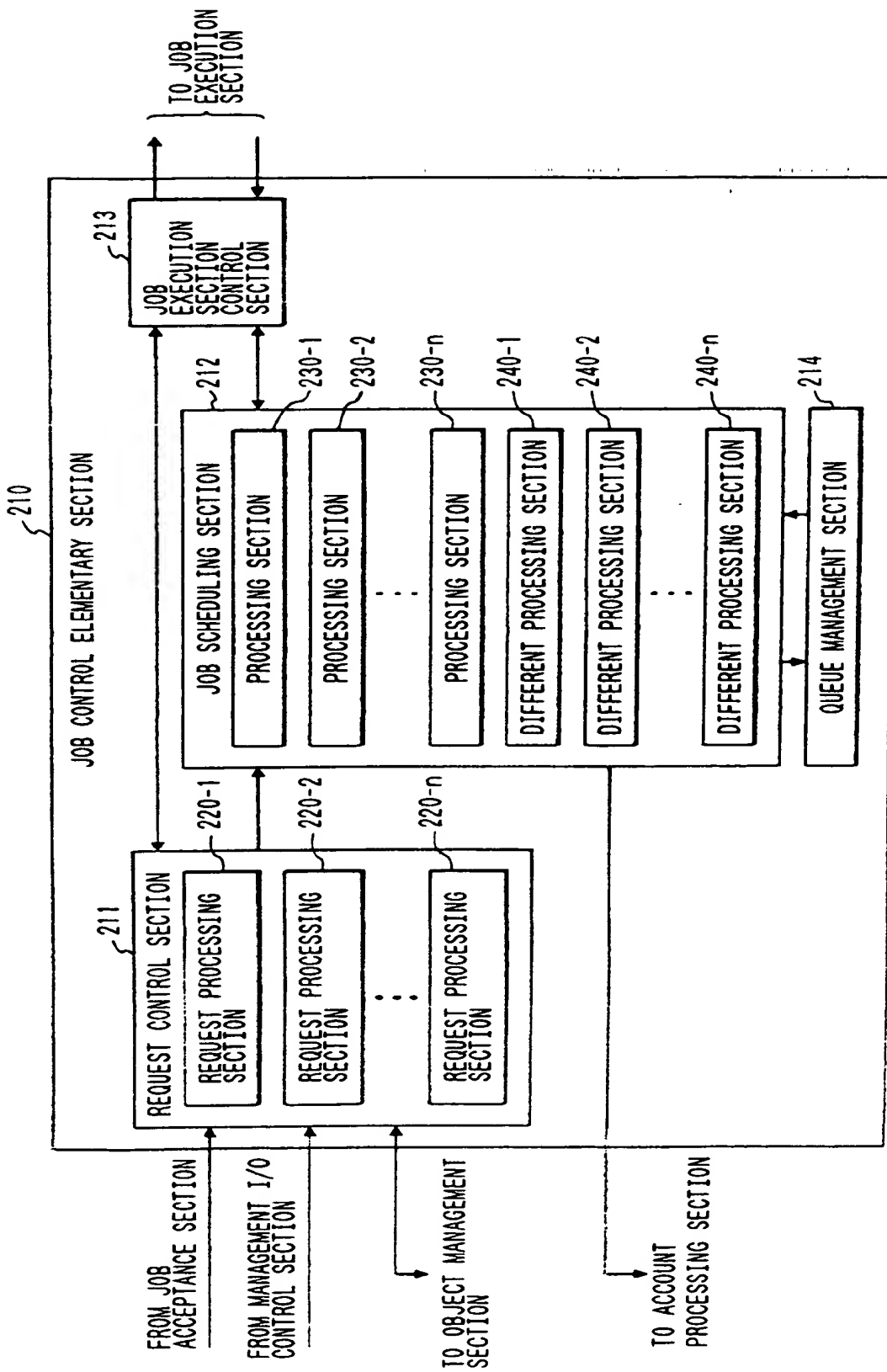


FIG. 29

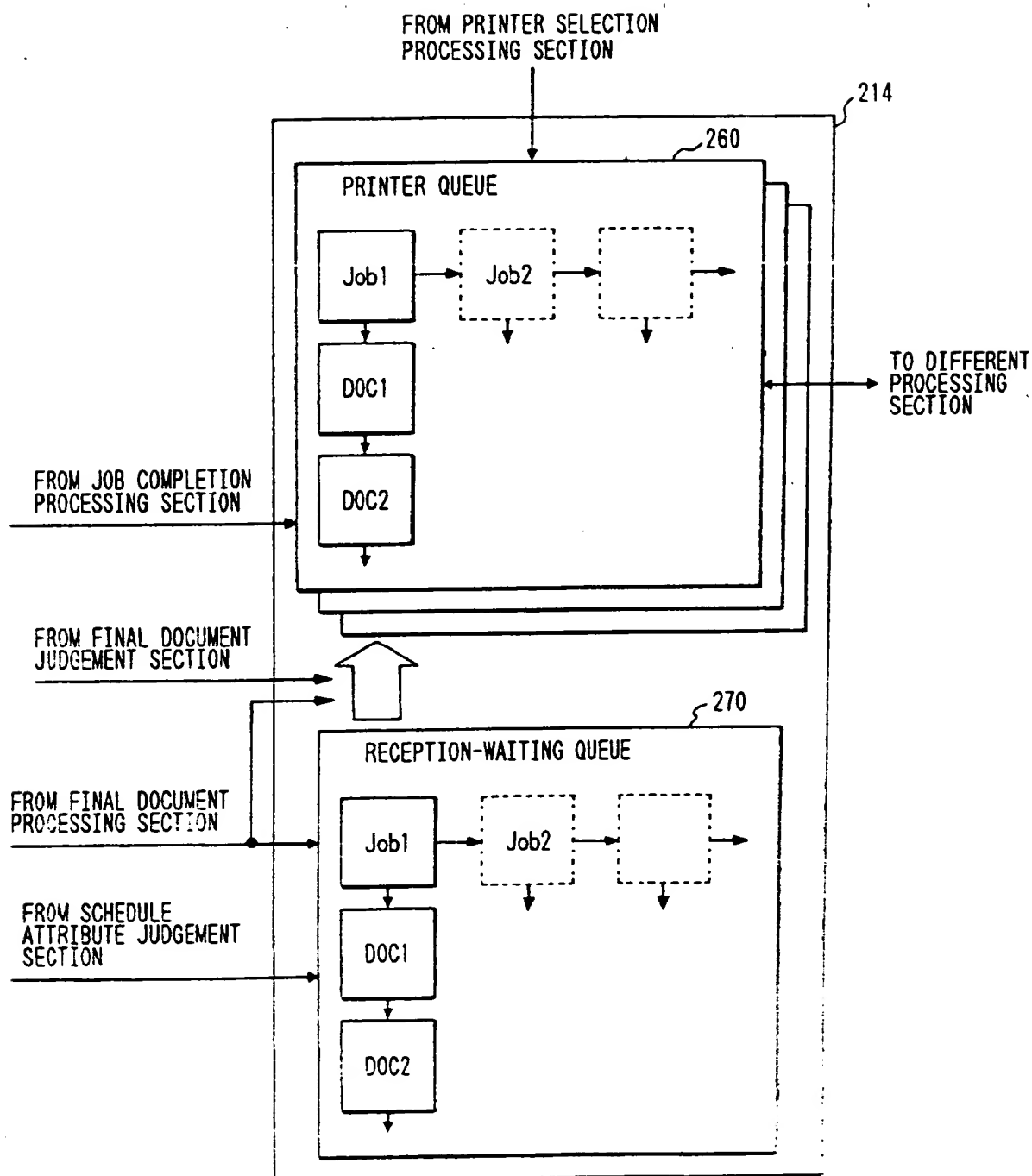


FIG. 30a

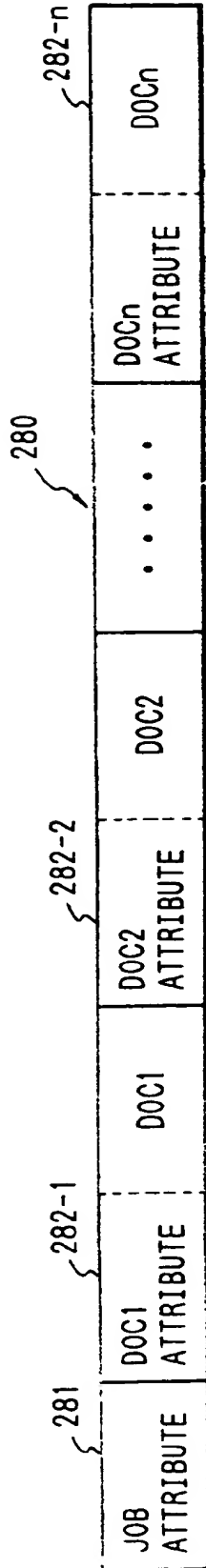


FIG. 30b

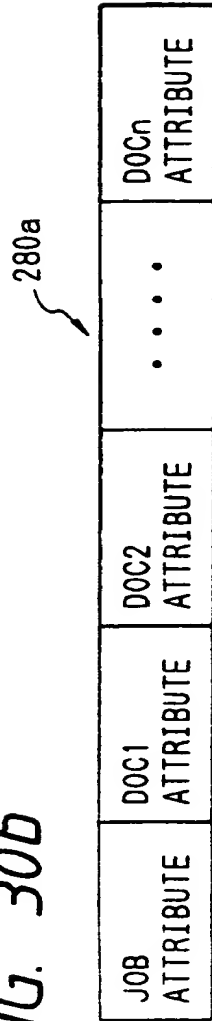


FIG. 30c

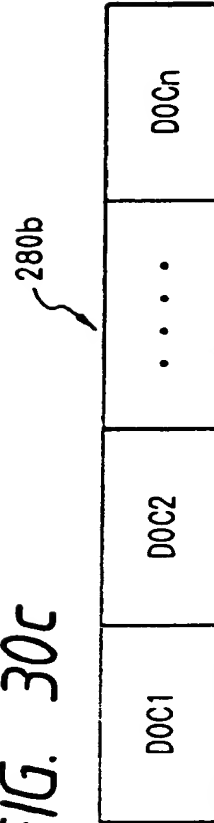


FIG. 31

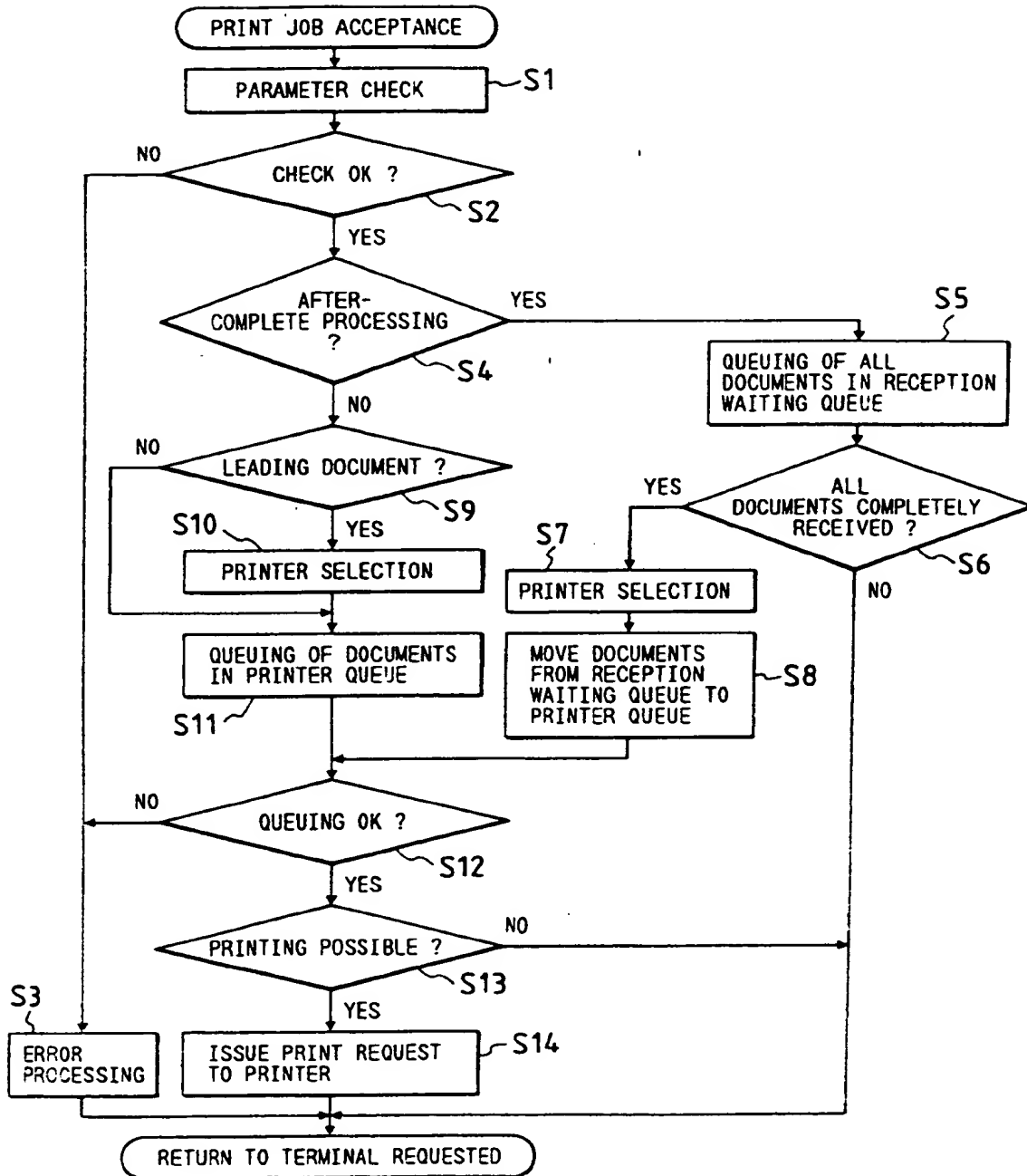


FIG. 32

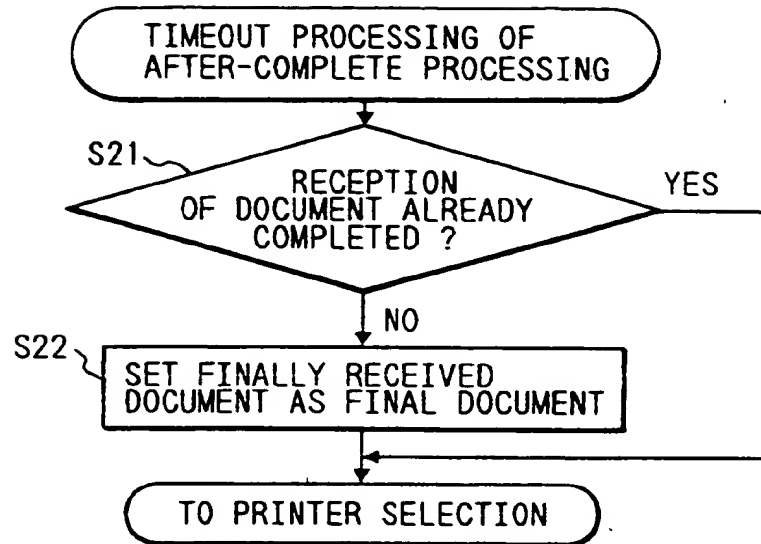


FIG. 33

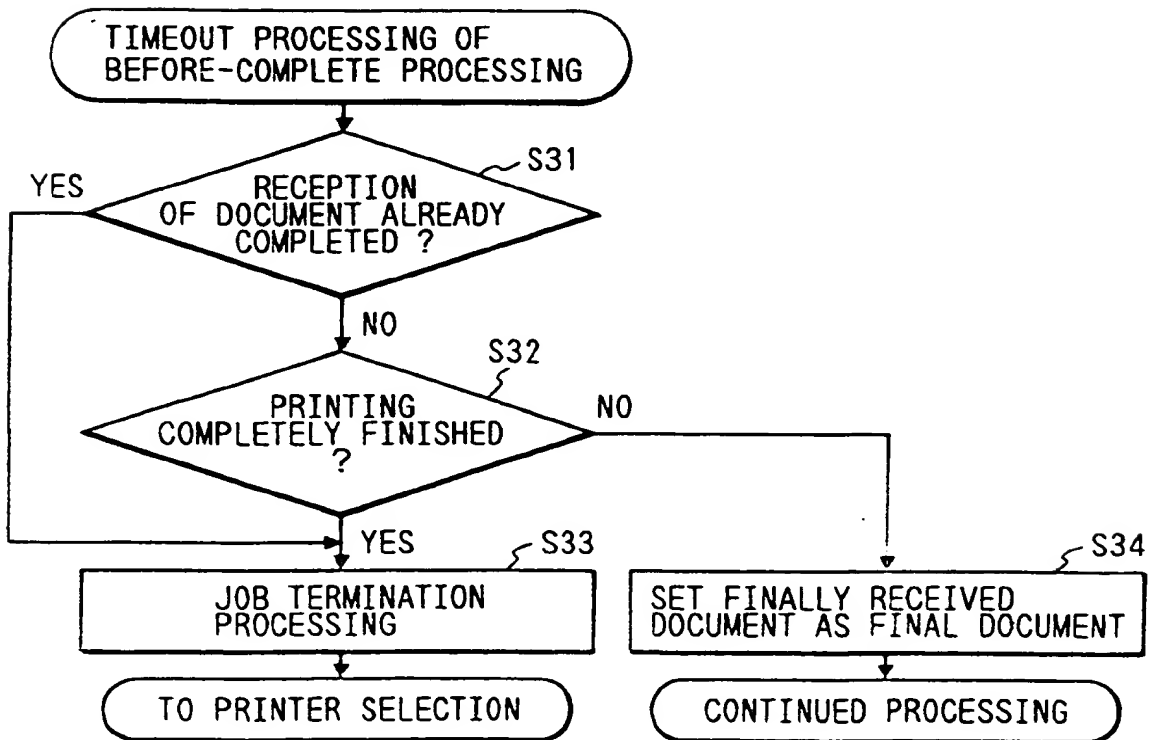


FIG. 34

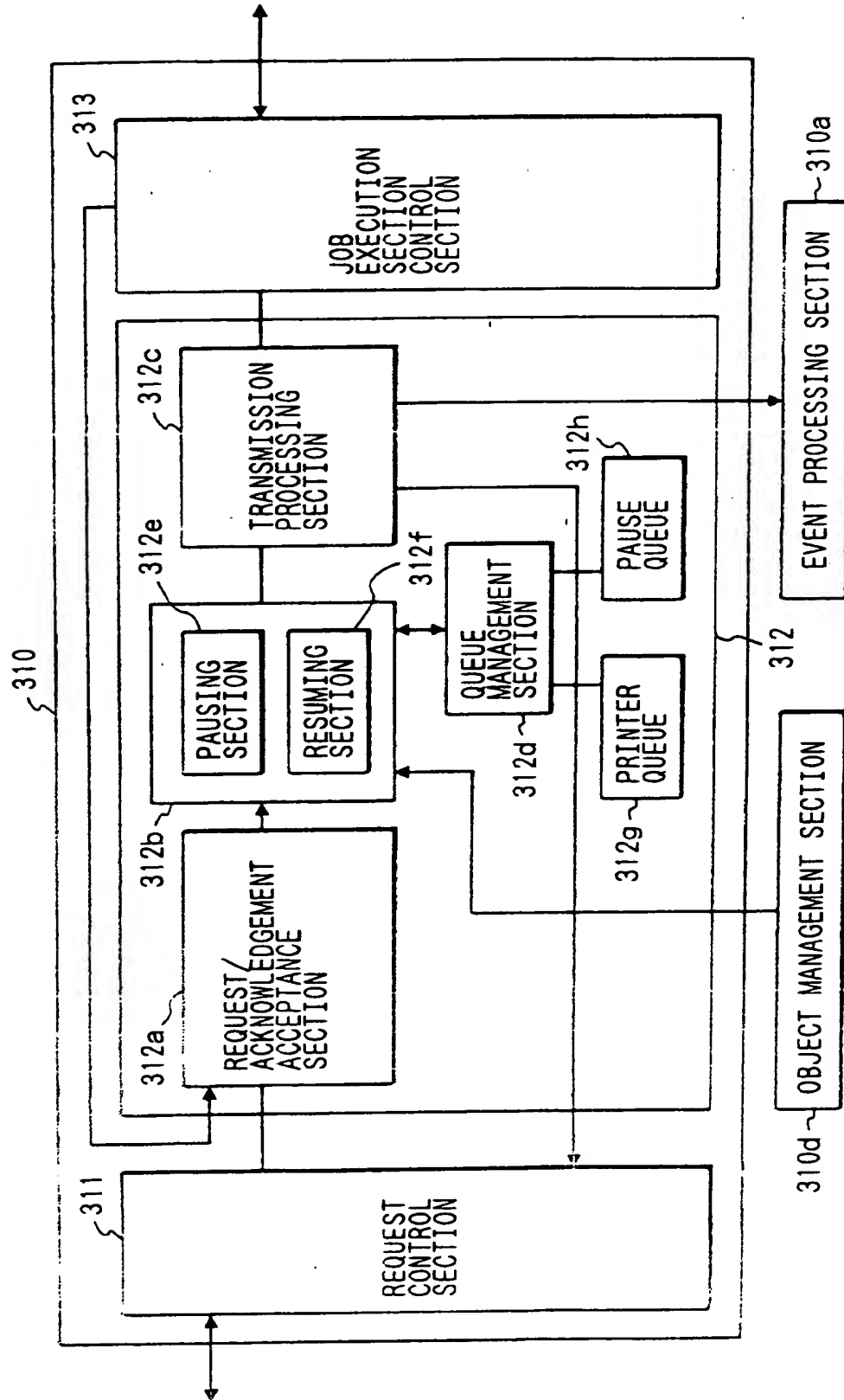


FIG. 35

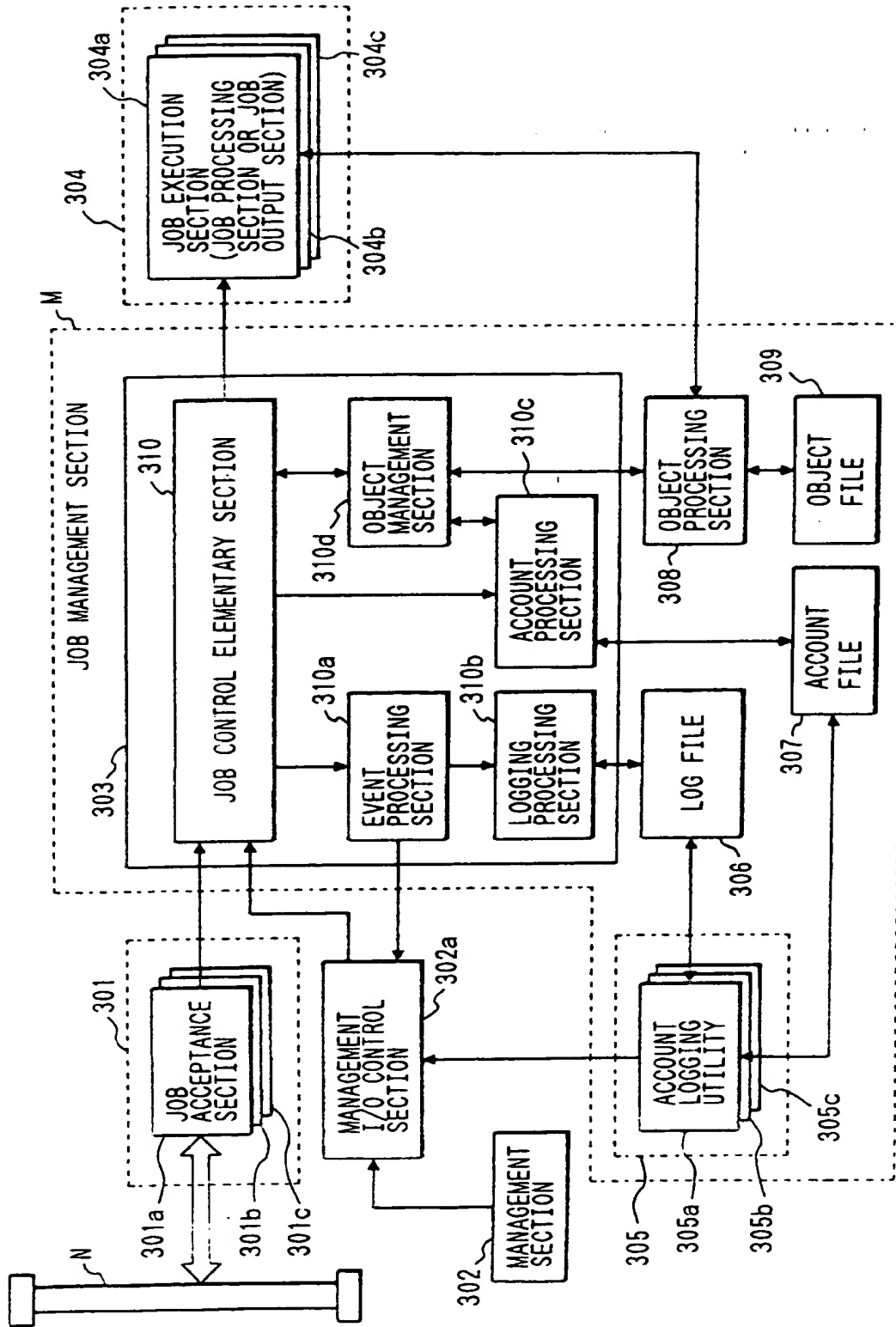


FIG. 36

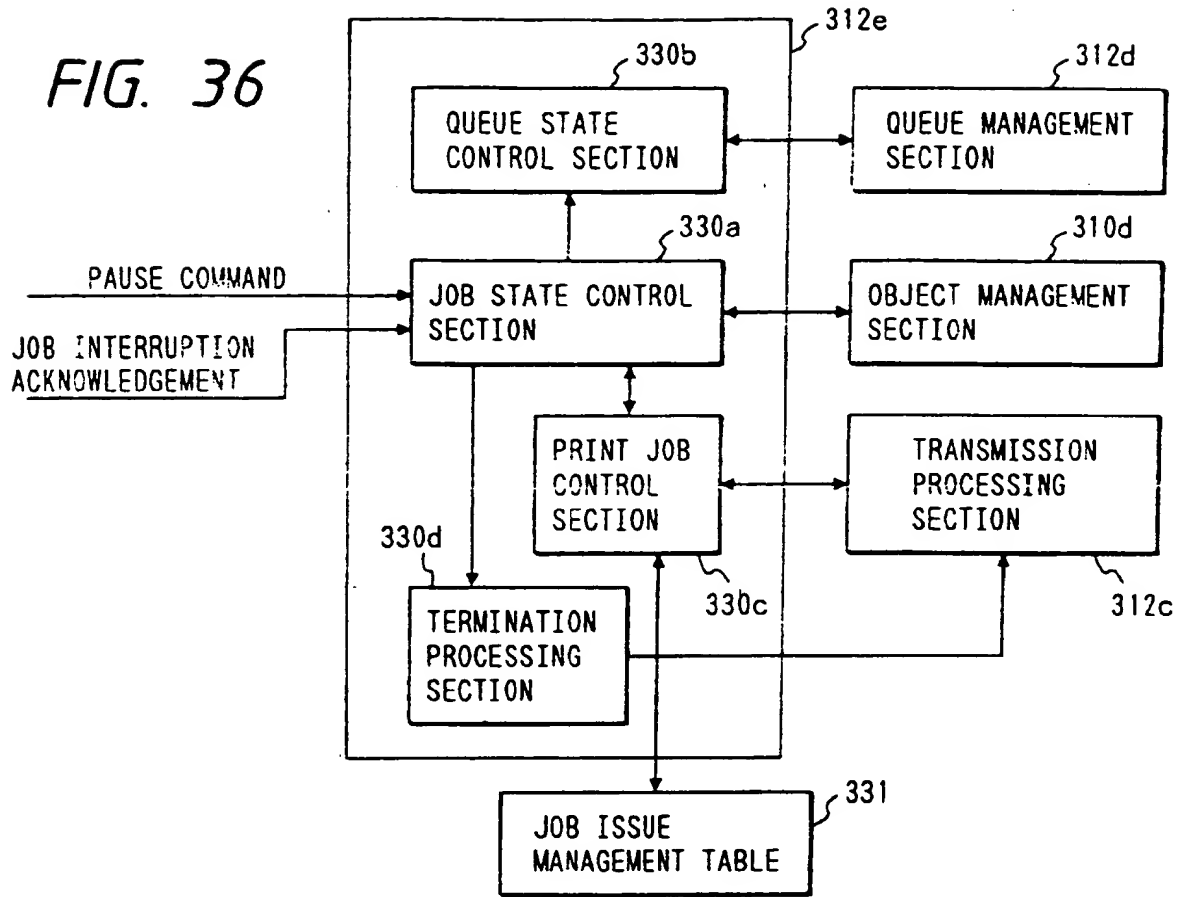


FIG. 38

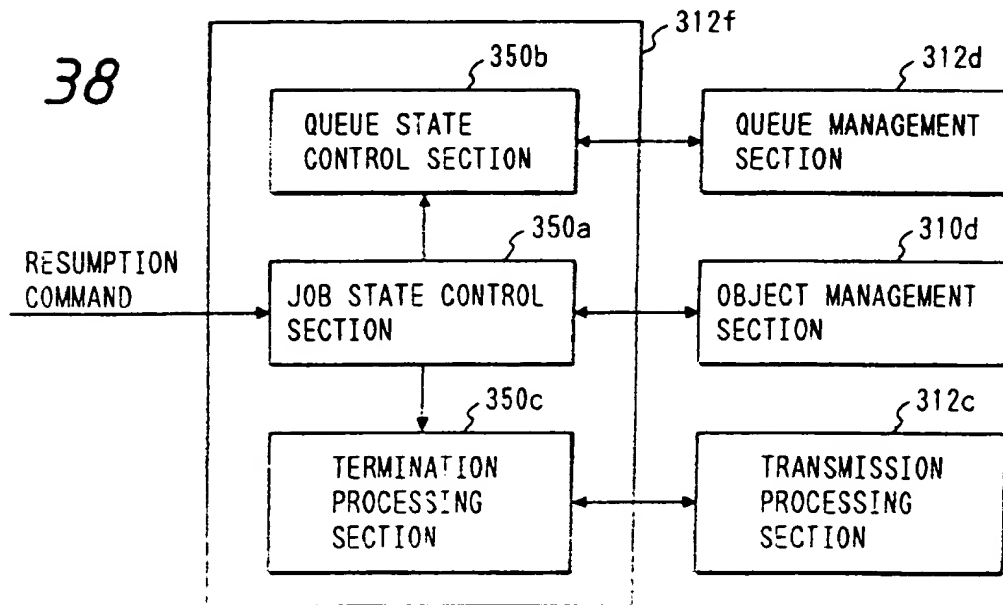


FIG. 37

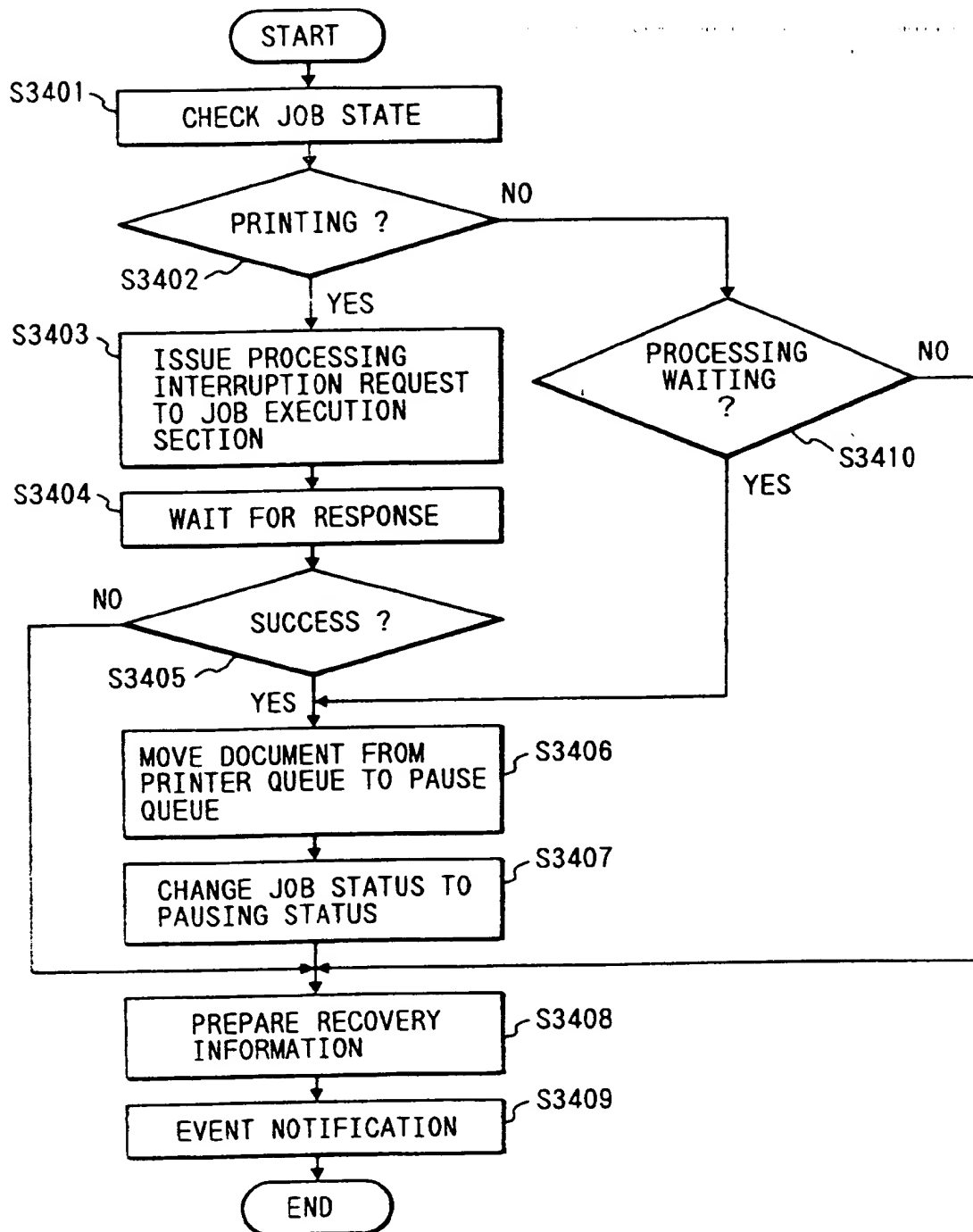


FIG. 39

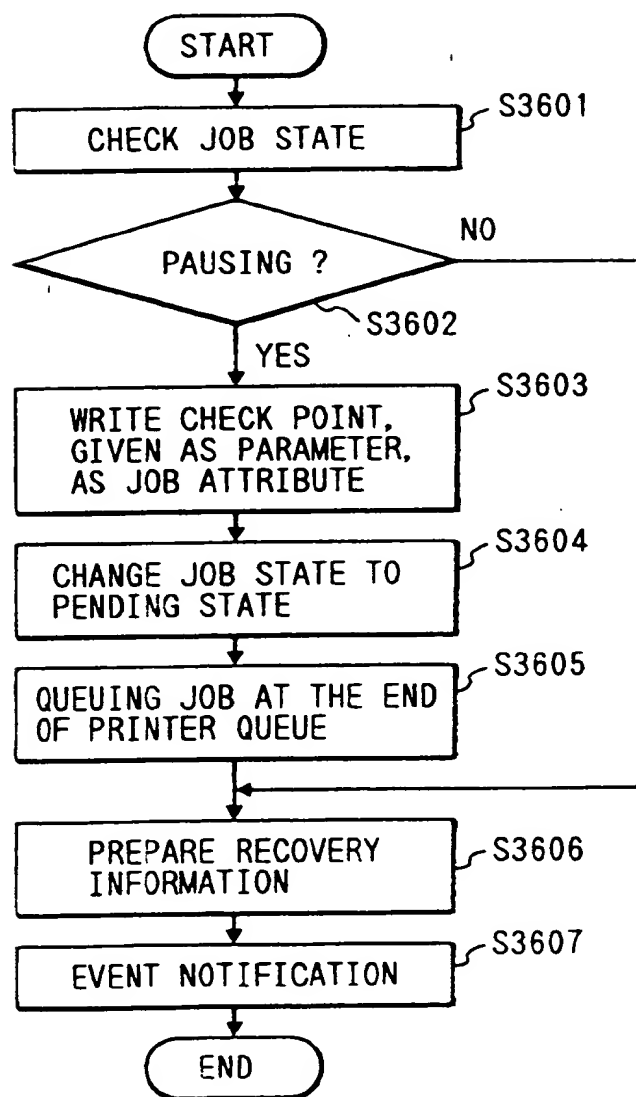


FIG. 40a

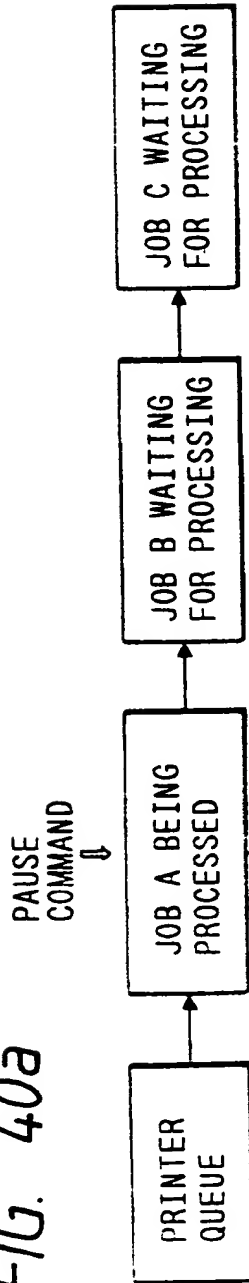


FIG. 40b

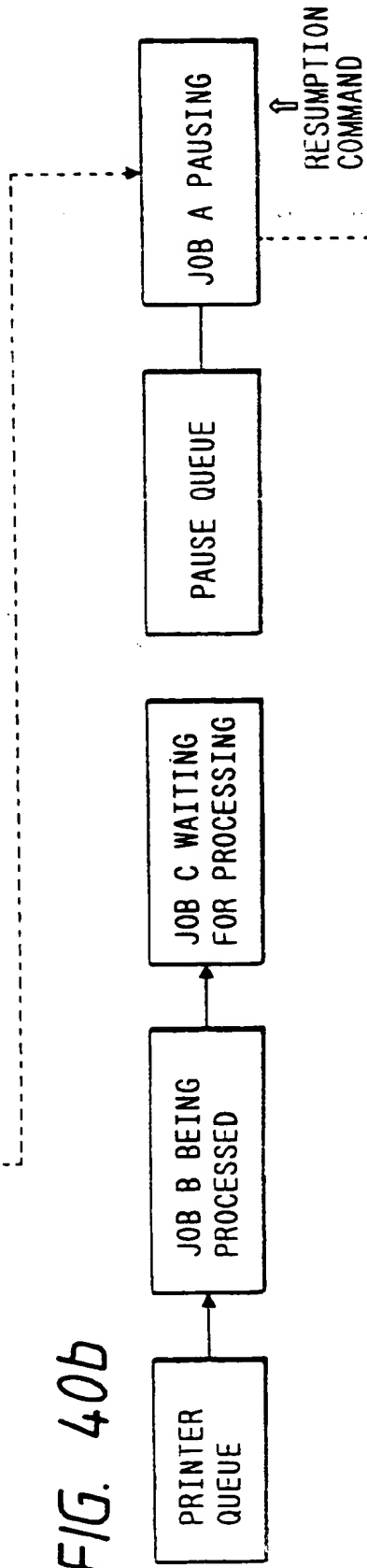
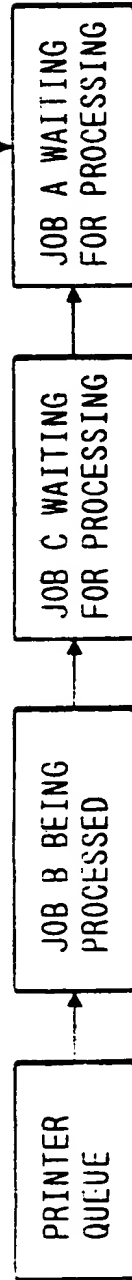


FIG. 40c



JOB EXECUTION
SECTION 4

JOB SCHEDULING
SECTION 12

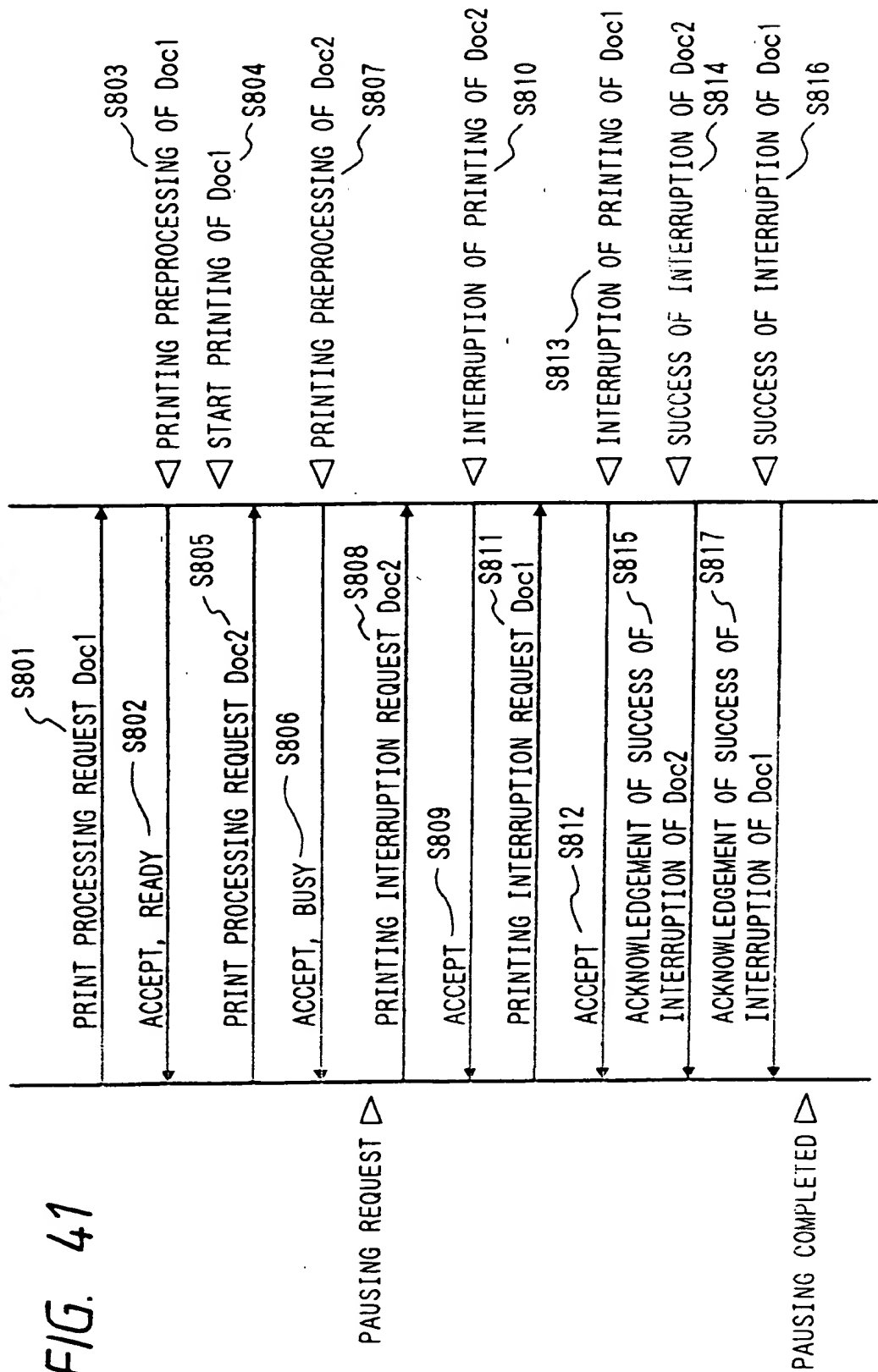


FIG. 42

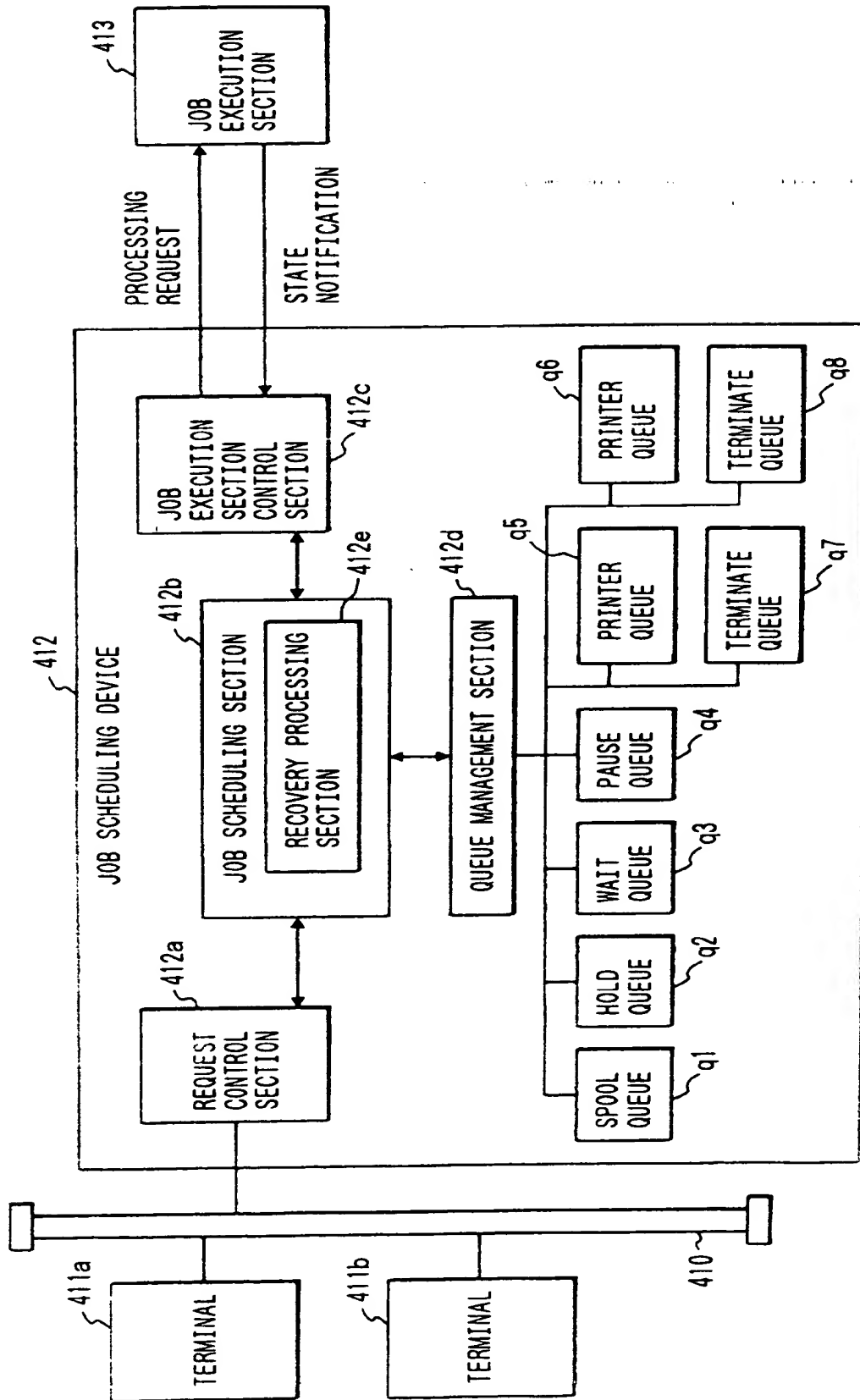


FIG. 43

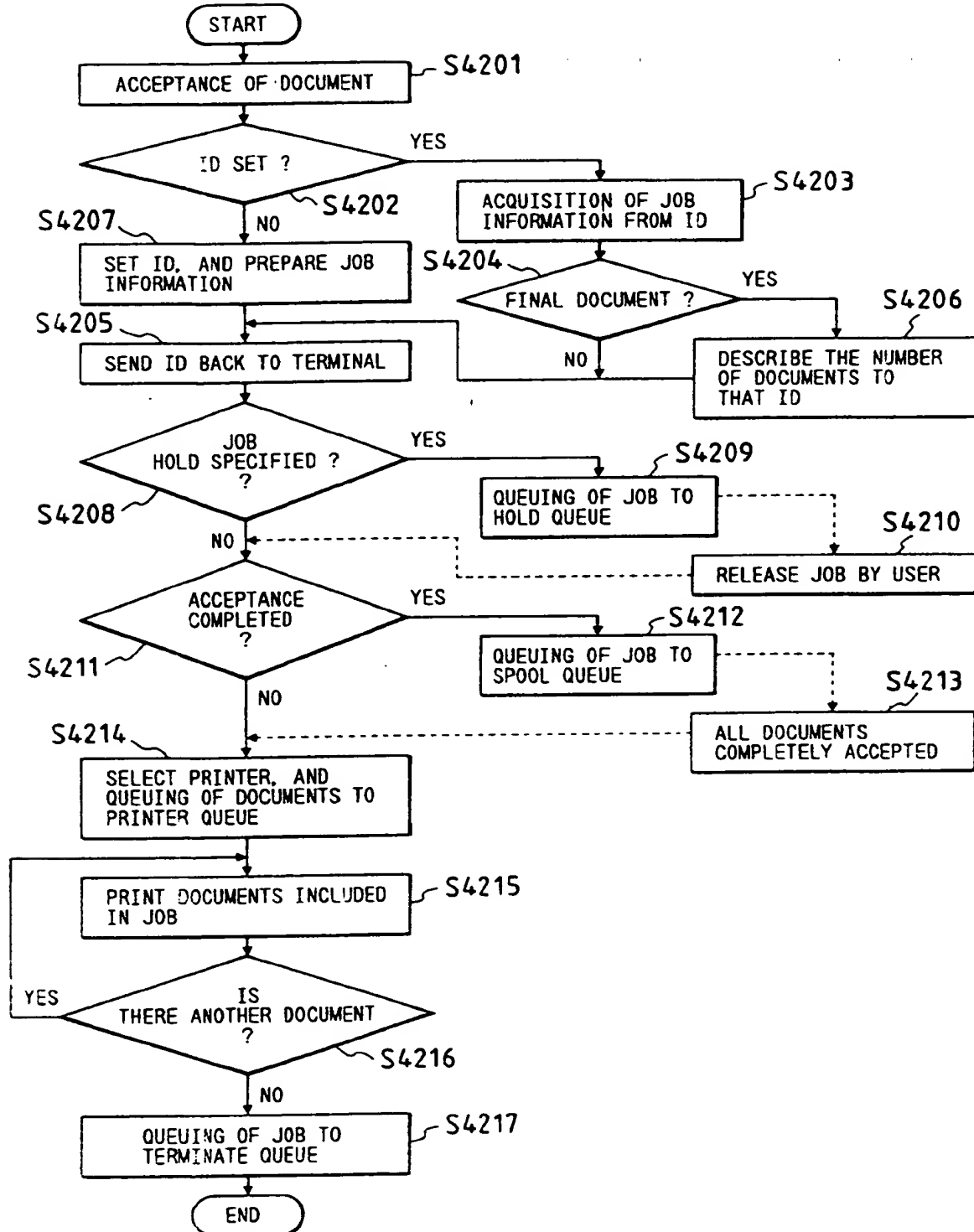


FIG. 44

MEMBER	EXPLANATION OF MEMBER
jobID	IDENTIFICATION NUMBER OF JOB
docsecNum	DOCUMENT IDENTIFICATION NUMBER OF ONE-JOB MULTIDOCUMENT
currentDocNum	docsecNum FOR ISSUING PRINT REQUEST
status	JOB STATUS
prevQID	PREVIOUSLY RETAINED QUEUE ID
jobCopyCount	NUMBER OF COPIES
requestCounter	NUMBER OF TIMES FOR REQUEST FOR PRINTING DOCUMENT TO JOB EXECUTION SECTION
completeCounter	NUMBER OF TIMES PROCESSING OF DOCUMENT IS TERMINATED
abortedCounter	NUMBER OF TIMES DOCUMENT BEING PROCESSED IS ABORTED
collateFlag	FLAG SHOWING WHETHER OR NOT COLLATION IS EFFECTED
resumeFlag	FLAG SHOWING WHETHER OR NOT RESUMING IS EFFECTED
previousStatus	PREVIOUS STATUS
retry	RETRY COUNTER WHEN REJECTED
complete	FLAG SHOWING WHETHER OR NOT THE DOCUMENT IS FINAL DOCUMENT
convertID	ID OF REQUESTED CONVERTER
ohPagesComp	NUMBER OF PAGES JOB IS OUTPUT
ohDocPagesComp	NUMBER OF PAGES DOCUMENT IS OUTPUT
ohCurJobState	CURRENT JOB STATE
ohPrevJobState	PREVIOUS JOB STATE
ohJobStateReason	REASON FOR CHANGE OF CURRENT JOB STATE
ohAssignJEID	ASSIGNED PRINTER ID
retryCounter	NUMBER OF RETRIES
processingFlag	FLAG SHOWING WHETHER OR NOT PRINTING IS BEING CARRIED OUT

430a

FIG. 45

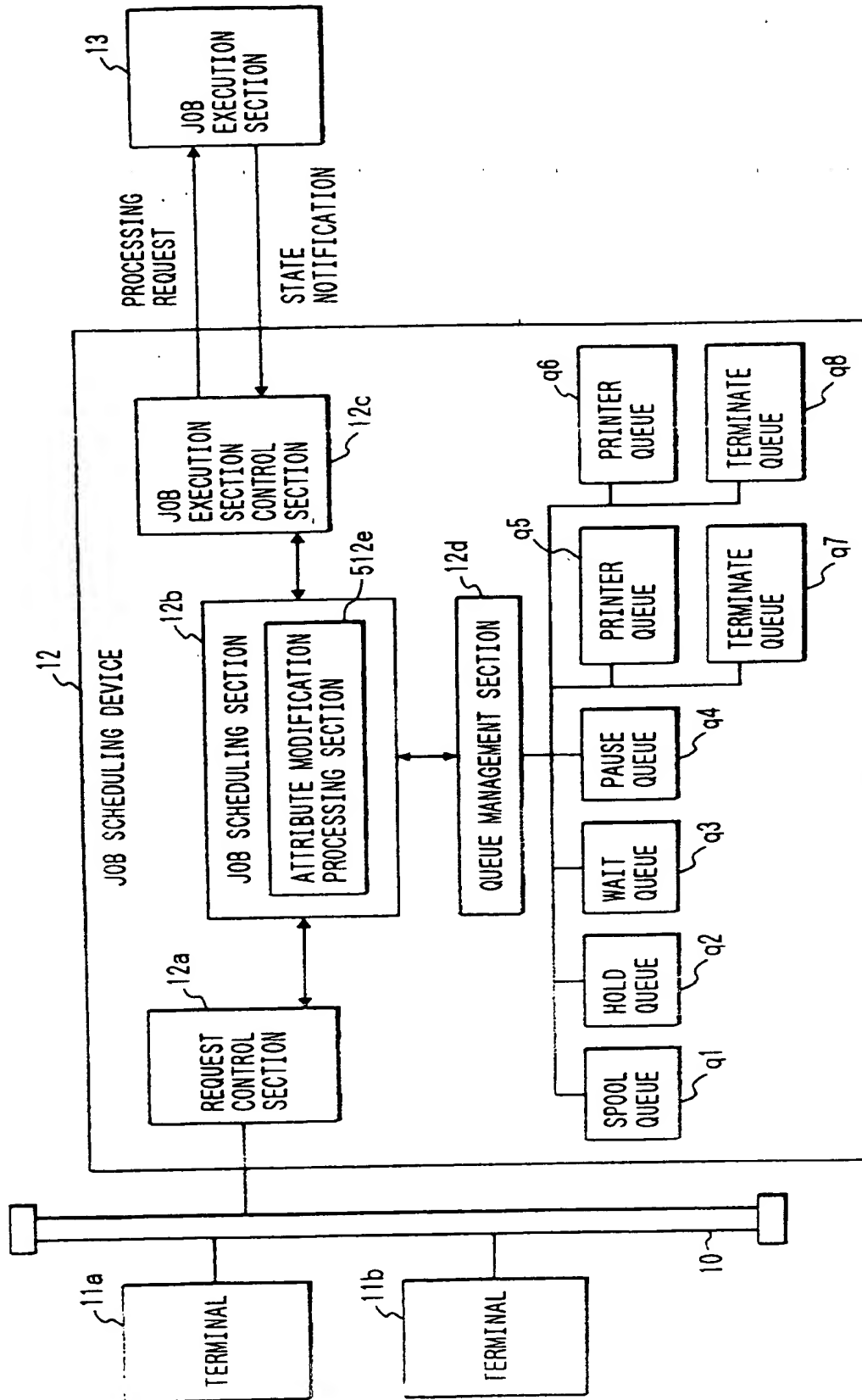


FIG. 46

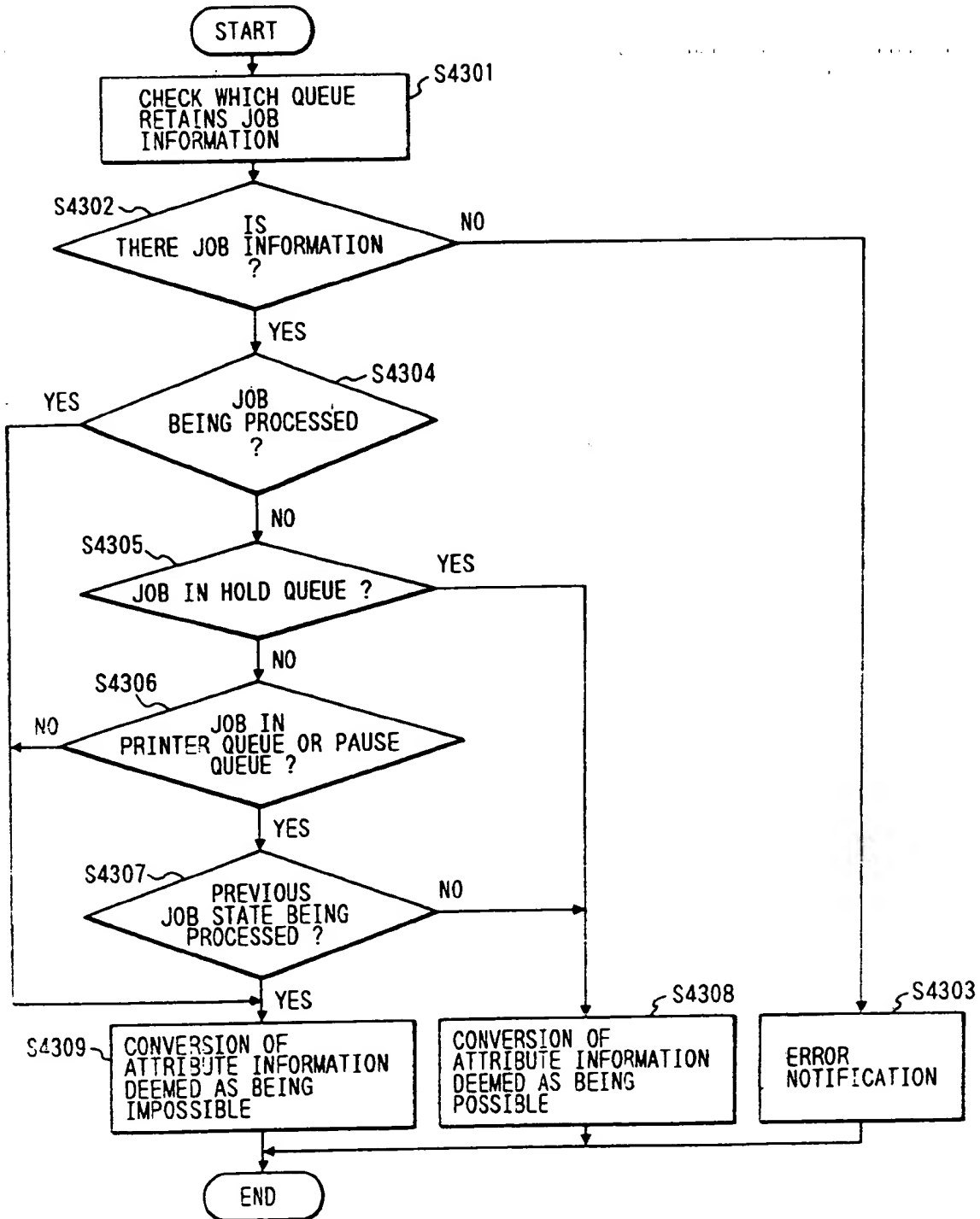


FIG. 47

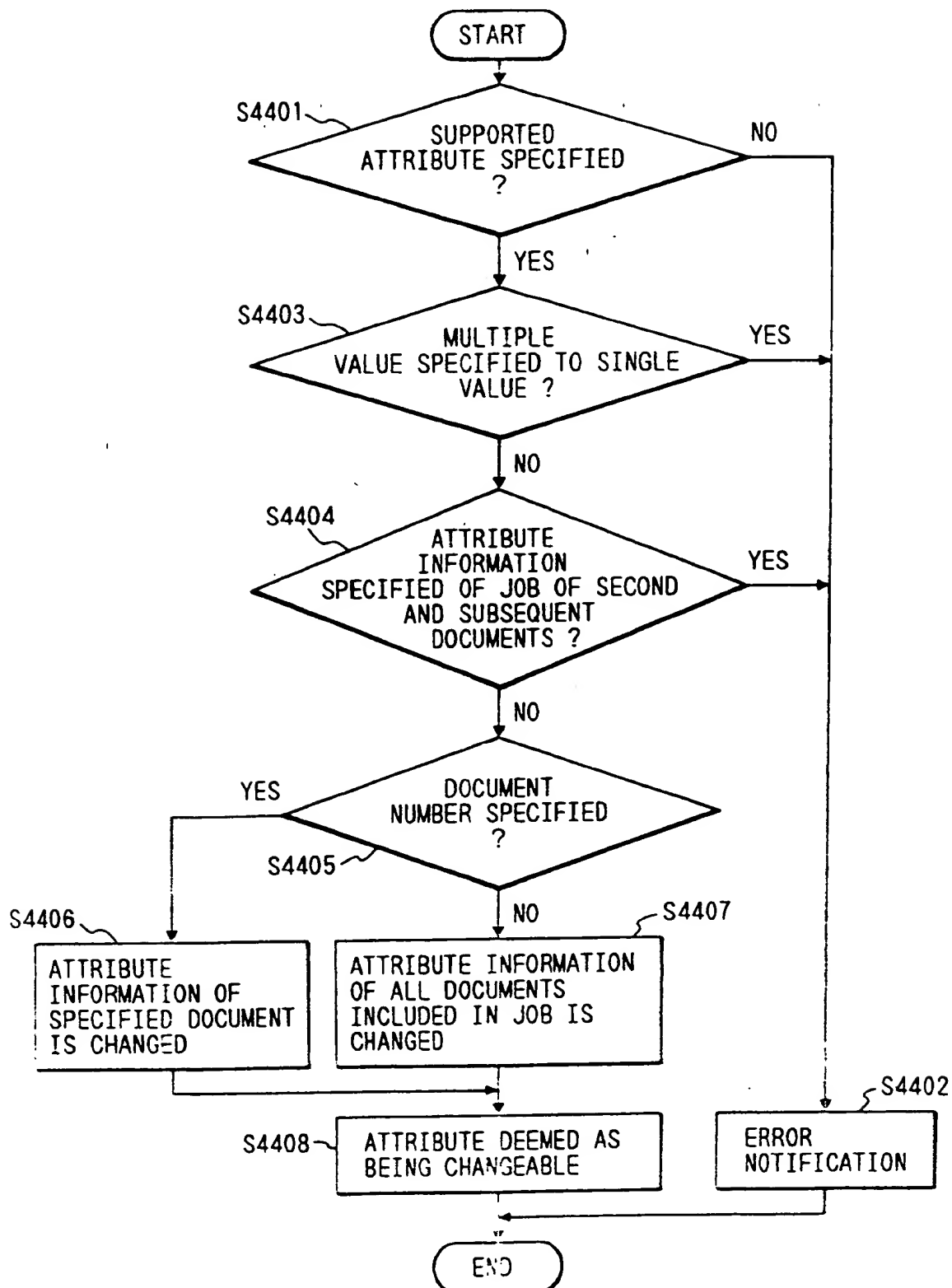


FIG. 48

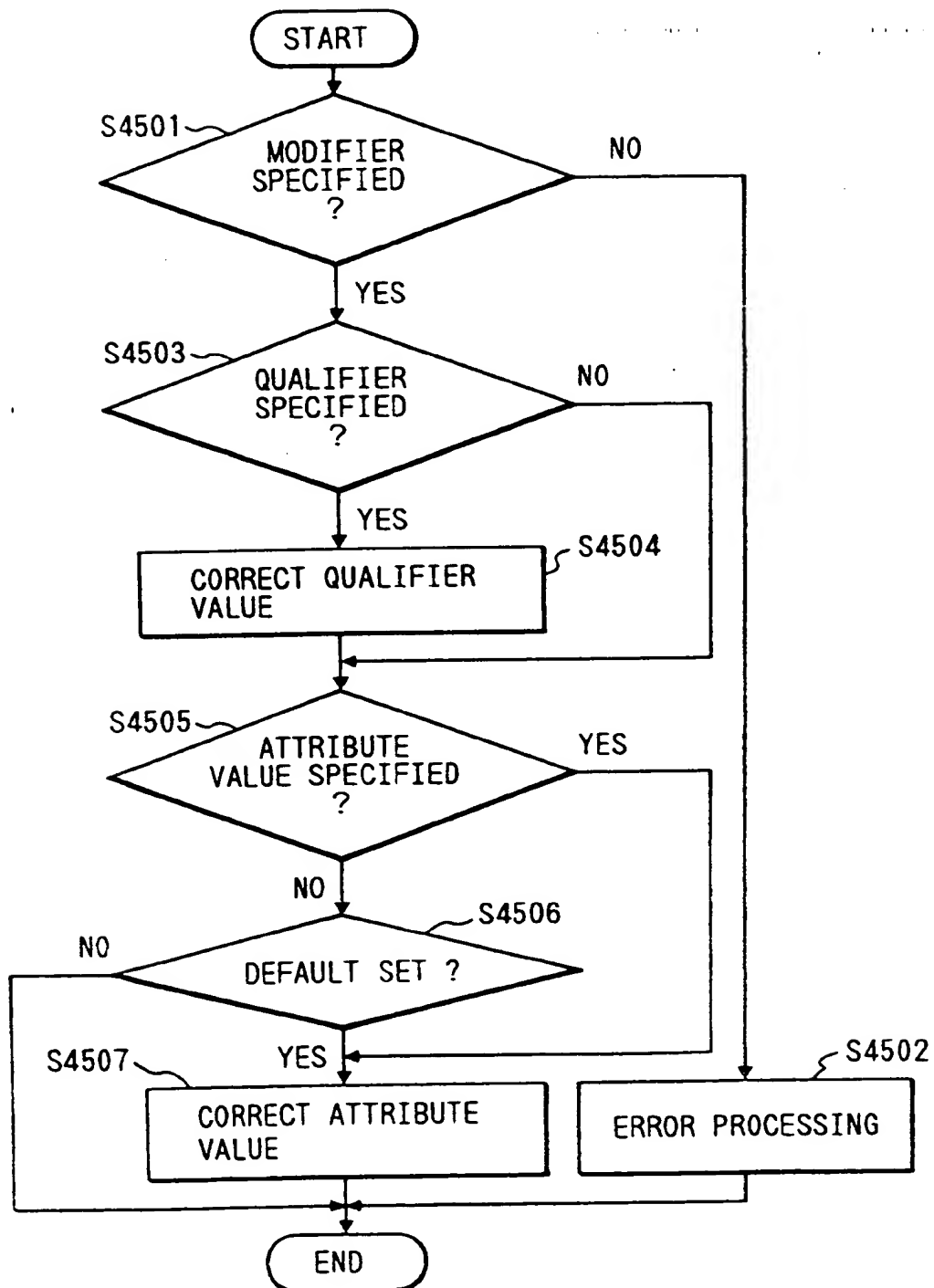


FIG. 49

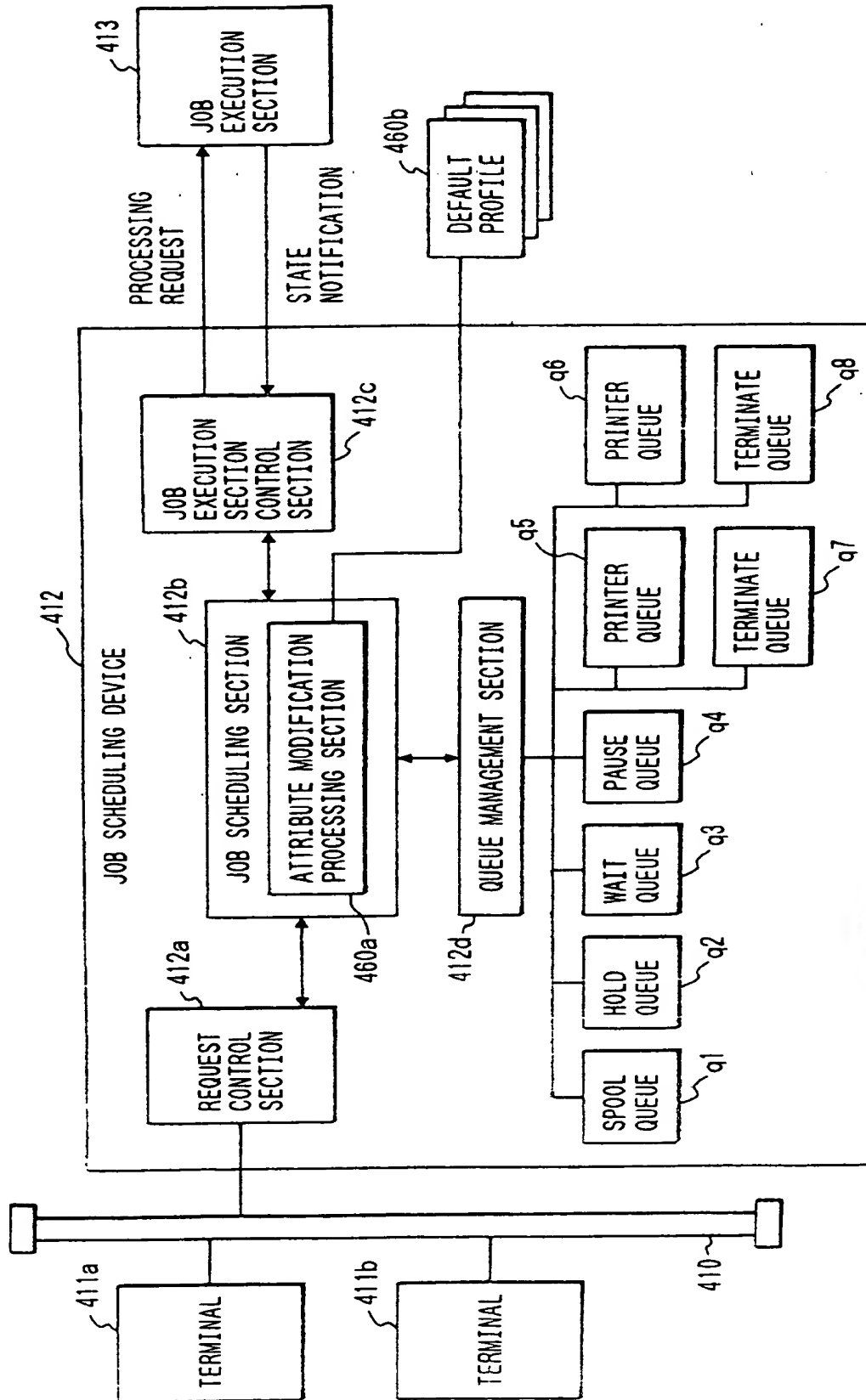


FIG. 50a

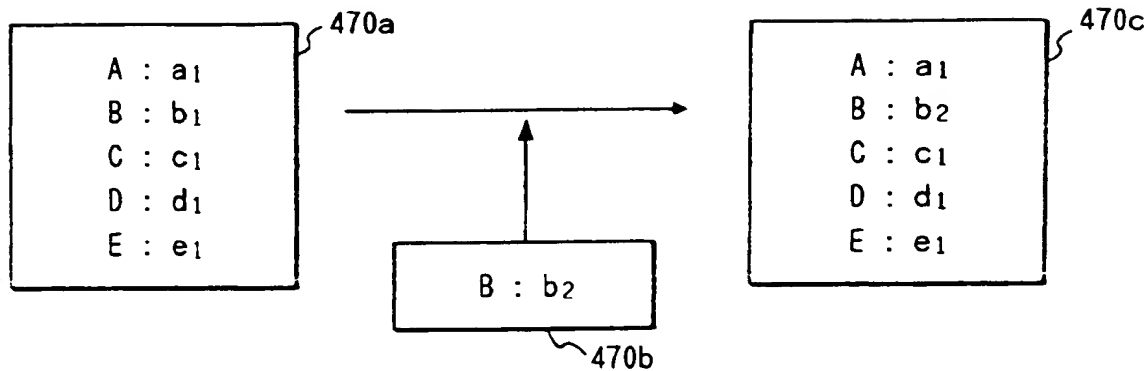


FIG. 50b

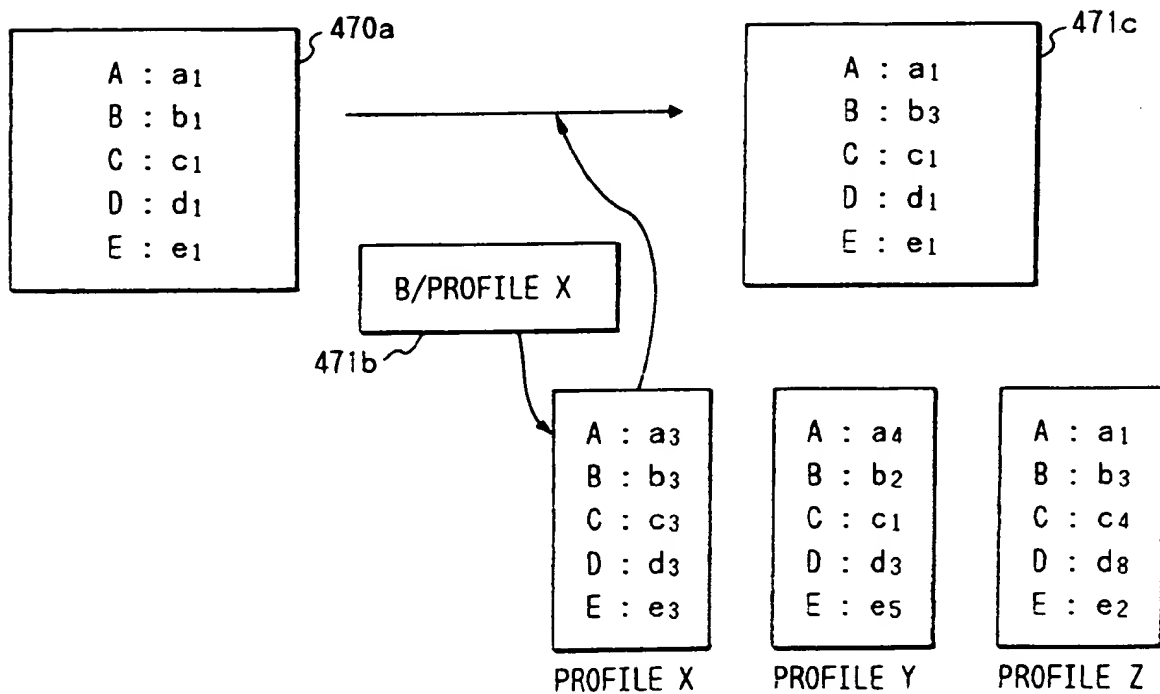
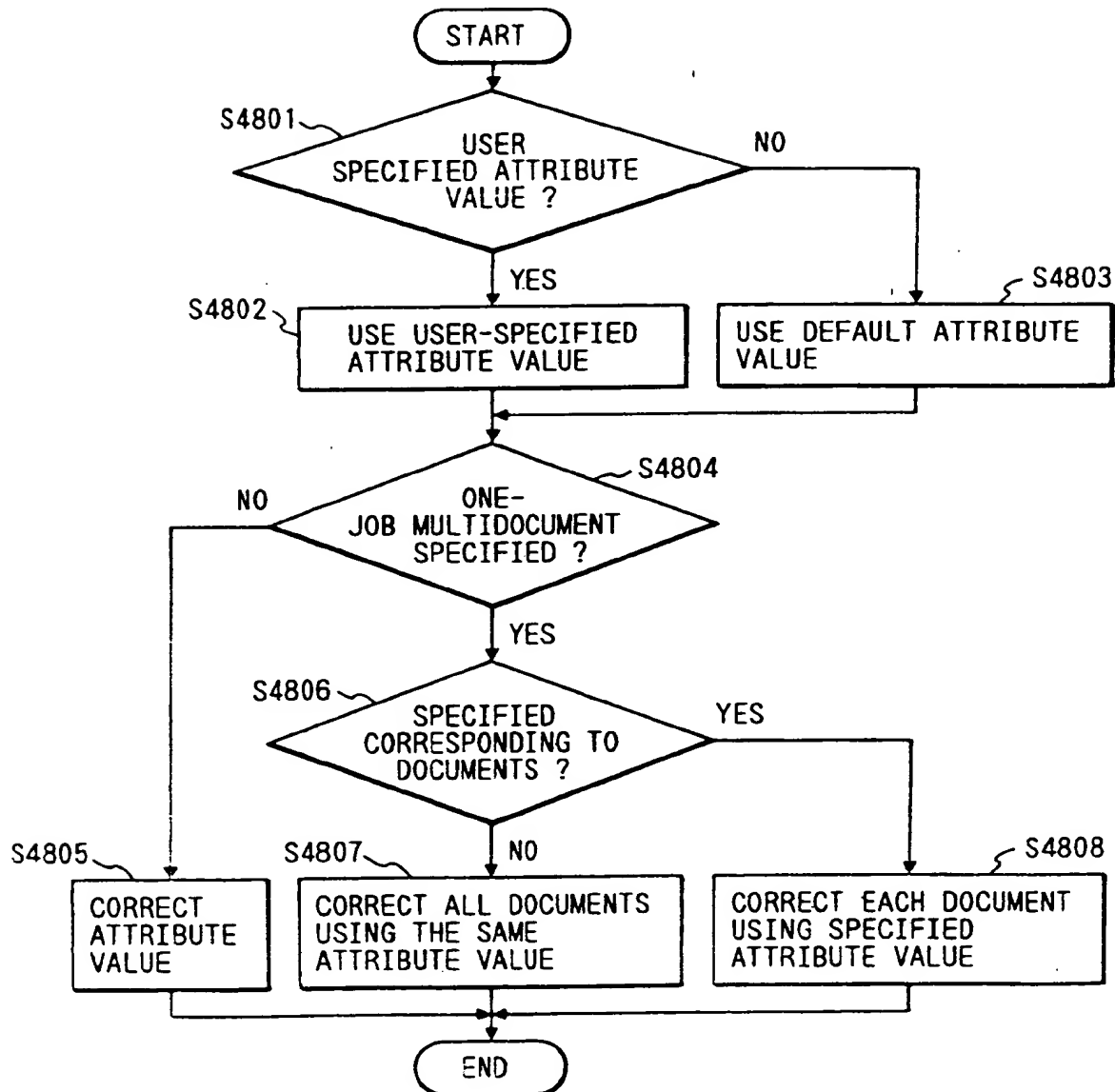
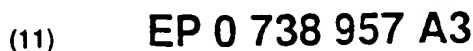


FIG. 51





(12)

(51) Int. Cl.⁶: G06F 3/12, G03G 15/00,
G06K 15/00

(21) Application number: 95116577.8

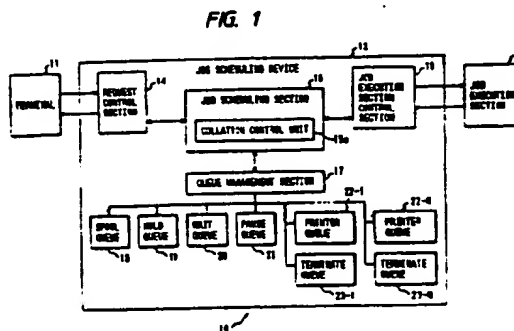
(22) Date of filing: 20.10.1995

- Yamada, Kentaro,
c/o Fuji Xerox Co., Ltd.
Takatsu-ku, Kawasaki-shi, Kanagawa (JP)
- Nishiyama, Koji,
c/o Fuji Xerox Co., Ltd.
Takatsu-ku, Kawasaki-shi, Kanagawa (JP)
- Nakatani, Tooru,
c/o Fuji Xerox Co., Ltd.
Takatsu-ku, Kawasaki-shi, Kanagawa (JP)
- Nakamura, Yoh,
c/o Fuji Xerox Co., Ltd.
Takatsu-ku, Kawasaki-shi, Kanagawa (JP)

**(74) Representative: Grünecker, Kinkeldey,
Stockmair & Schwanhäusser
Anwaltssozietät
Maximilianstrasse 58
80538 München (DE)**

(72) Inventors:
• Suzuki, Akihiro,
c/o Fuji Xerox Co., Ltd.
Takatsu-ku, Kawasaki-shi, Kanagawa (JP)

(57) A user equipment issues a job for a plurality of documents to a printing system by one operation. The user equipment assigns attributes to the job, the attributes including the specification of collation/uncollation, the designation of processing start instruction wait, the designation of exclusive processing, and the designation of password input wait. The printing system effects printing of the plurality of documents corresponding to attributes of the documents under control of a job scheduling device. The job scheduling device carries out the pausing of documents included in the job, the modification of attributes, addressing of document receiving failures, and search of a document whose format is to be converted, thereby making it possible to efficiently print the plurality of documents.





European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 11 6577

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. CL. 6)
X	US-A-5 179 637 (F. M. NARDOZZI)	1	G06F3/12 G03G15/00 G06K15/00
Y	* the whole document *	2	
X	EP-A-0 529 808 (XEROX CORPORATION)	1	
A	* column 1, line 54 - column 3, line 51 *	2	
	* column 5, line 37 - column 6, line 6 *		
Y	EP-A-0 588 513 (XEROX CORPORATION)	1,2	
	* column 4, line 40 - column 6, line 52 *		
Y	EP-A-0 479 494 (XEROX CORPORATION)	1,2	
	* *		
Y	DE-A-44 08 355 (RICOH CO., LTD.)	2	
	* column 12, line 16 - line 67 *		
			TECHNICAL FIELDS SEARCHED (Int. CL. 6)
			G06F
Place of search		Date of completion of the search	Examiner
BERLIN		15 November 1996	Abram, R
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons</p> <p>& : member of the same patent family, corresponding document</p>			



European Patent
Office

EP 95116577

CLAIMS INCURRING FEES

The present European patent application comprised at the time of filing more than ten claims.

- ☐ All claims fees have been paid within the prescribed time limit. The present European search report has been drawn up for all claims.
- ☐ Only part of the claims fees have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims and for those claims for which claims fees have been paid, namely claims:
- ☐ No claims fees have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims.

X LACK OF UNITY OF INVENTION

The Search Division considers that the present European patent application does not comply with the requirement of unity of invention and relates to several inventions or groups of inventions, namely:

see attachment

- ☐ All further search fees have been paid within the fixed time limit. The present European search report has been drawn up for all claims.
- ☐ Only part of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the inventions in respects of which search fees have been paid, namely claims:
- ☒ None of the further search fees has been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the invention first mentioned in the claims, namely claims: 1,2

ATTACHMENT

EP 951 16577

1. Claims 1,2:
Job processing system for multi-document jobs; outputting and collating multiple copies.
2. Claims 3,4:
Job processing system; specifying wait times for documents/jobs.
3. Claim 5:
Job processing system; preventing job processing unless password specified.
4. Claim 6:
Job processing system; informing terminal of job wait state.
5. Claims 7,8:
Print processor with job format conversion.
6. Claims 9-12:
Print processor and processing method for cutting short multi-document job if documents fail to arrive quickly enough (e.g. in case of receiving failure).
7. Claims 13,14:
Job processing device: moving interrupted jobs to a second queue, and returning them to the first queue on resumption.
8. Claims 15,16:
Multi-queue job scheduling device; recovery from failures during scheduling; changing job attributes.